AEDC-TR-77-89

SIMPLIFIED INPUT FOR CERTAIN AERODYNAMIC NOSE CONFIGURATIONS TO THE GRUMMAN QUICK-GEOMETRY SYSTEM (A KWIKNOSE USER'S MANUAL)

Frederick L. Shope
ARO, Inc., a Sverdrup Corporation Company

VON KÁRMÁN GAS DYNAMICS FACILITY
ARNOLD ENGINEERING DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
ARNOLD AIR FORCE STATION, TENNESSEE 37389

February 1978

Final Report for Period January to September 1977

Approved for public release; distribution unlimited.

Prepared for

ARNOLD ENGINEERING DEVELOPMENT CENTER/DOTR
ARNOLD AIR FORCE STATION, TENNESSEE 37389

NOTICES

When U. S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Qualified users may obtain copies of this report from the Defense Documentation Center.

References to named commerical products in this report are not to be considered in any sense as an indorsement of the product by the United States Air Force or the Government.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

APPROVAL STATEMENT

This report has been reviewed and approved.

Elton R Thompson

ELTON R. THOMPSON
Project Manager, Research Division
Directorate of Test Engineering

Approved for publication:

FOR THE COMMANDER

Marion L. Laster

MARION L. LASTER
Director of Test Engineering
Deputy for Operations

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
AEDC-TR-77-89

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*
SIMPLIFIED INPUT FOR CERTAIN AERODYNAMIC NOSE CONFIGURATIONS TO THE GRUMMAN QUICK-GEOMETRY SYSTEM (A KWIKNOSE USER'S MANUAL)

5. TYPE OF REPORT & PERIOD COVERED
Final Report - January to September 1977

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Frederick L. Shope, ARO, Inc.

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Arnold Engineering Development Center/DOT
Air Force Systems Command
Arnold Air Force Station, Tennessee 37389

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Program Element 65807F

11. CONTROLLING OFFICE NAME AND ADDRESS
Arnold Engineering Development Center/DOS
Arnold Air Force Station, Tennessee 37389

12. REPORT DATE
February 1978

13. NUMBER OF PAGES
93

14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*

15. SECURITY CLASS. *(of this report)*
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

Available in DDC

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

handbook
user needs
input
Fortran
computer programs
geometric forms
mathematical model
flow fields
aerodynamic configurations

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report is a user's manual for a Fortran computer program KWIKNOSE which, for certain axisymmetric and nonaxisymmetric nose configurations, provides simplified geometric input to the Grumman QUICK-geometry system, which in turn provides geometric information to various numerical flow codes. For a wide variety in choice of input parameters, KWIKNOSE sets up the QUICK input for an arbitrary sequence of conical and ogival sections. In this process, KWIKNOSE

20. ABSTRACT (Continued)

performs the tedious computations necessary to locate the intersection points of successive arcs and to insert optional fillets or rounds over nontangent intersections. In addition, the code is capable of inserting arbitrary multiple slicing planes into the top, bottom, and side of the vehicle. Slicing plane intersections may be filleted or rounded. Thus, for a minimum of input and manual calculation by the user, KWIKNOSE is tailored to modeling the geometry of a sliced multiconic vehicle capped with an asymmetrically ablated nose. This manual provides check cases for the various geometry options, a description of input and output, and a listing of the source deck.

PREFACE

The work reported herein was conducted by the Arnold Engineering Development Center (AEDC), Air Force Systems Command (AFSC), under program element 65807F. The results were obtained by ARO, Inc., AEDC Division (a Sverdrup Corporation Company), operating contractor for the AEDC, AFSC, Arnold Air Force Station, Tennessee. The research was done under ARO Project No. V33A-A8A. Elton R. Thompson was the Air Force project manager. The manuscript was submitted for publication on September 2, 1977.

CONTENTS

ILLUSTRATIONS

TABLES

TABLES (continued) Page

1.0 INTRODUCTION

This document describes a computer program which allows greatly simplified input, for certain common axisymmetric and nonaxisymmetric nose-vehicle geometries of general interest, to the more general Grumman QUICK-geometry system. QUICK, a Fortran computer program developed by Vachris and Yaeger (Refs. 1 and 2), provides the user with a simple and well organized method of constructing and interrogating a mathematical model of an arbitrary, three-dimensional flight vehicle. This geometry system has primary application to the numerical computation of the flow fields over arbitrary aerodynamic configurations. QUICK was developed in two phases, the QUICKDEF phase, which provides facilities to build and check the geometry model, and the QUICKLOK phase, which provides surface coordinates and derivatives to user programs without requiring of the user a detailed knowledge of how the model was constructed. Current applications of the QUICKLOK phase include the time-dependent blunt body program BLUNT (Ref. 3); the supersonic, three-dimensional, external, inviscid flow-field code STEIN (Refs. 3 and 4); and the multishocked, three-dimensional, supersonic finite difference method of Kutler, Reinhardt, and Warming (Ref. 5). The QUICKDEF phase, of primary concern here, is a highly user-oriented program which greatly simplifies the geometric input by providing general curve fitting services to define arbitrary bodies from a minimum of logical and numerical input. In essence, QUICK requires the user to define each unique cross section of the vehicle in a plane normal to the coordinate axis of the vehicle by merely identifying the types of curves (lines, ellipses, parabolas, ...) which form the segmented cross section. The cross sections then become numerically specific when the user further defines, logically and numerically, certain key control points which determine the intersections and slopes of the curves comprising the cross sections. The only two limitations imposed by QUICK upon the vehicle geometry are (1) that the vehicle must have a vertical plane of symmetry and (2) that each cross section must be definable in terms of a single-valued radius vector swung about a single point within the cross section. While these limitations prevent complete treatment of certain vehicle geometries (such as engine inlets, the oblique wing supersonic transport, or the double vertical stabilizer of the F-15), QUICK is an important advance in the description of vehicle geometry and makes practical the solution of flows over such complicated bodies as the space shuttle orbiter.

In contrast, QUICK is also well suited to the treatment of relatively simple bodies such as sphere-cones, biconics, and slab deltas, which are of basic interest to aerodynamicists. It was therefore deemed logical to develop yet another sequential computer program to set up input to QUICK for certain such simple geometries, thus reducing user-prepared input to a matter of two or three cards for the programmed geometries. This fourth program PREKWIK is documented in a previous user's manual.

During the development of PREKWIK, the need was recognized to study more complicated nose shapes than the simple spherical cap treated by PREKWIK. Interest centered around the effects of ablated nose shapes (Fig. 1) upon the aerodynamics of otherwise simple bodies such as cones and biconics with slices on the conic portions of the vehicle. A feature which most of the symmetrical noses have in common is that they can all be modeled (with varying degrees of accuracy) as a stack of conical and ogival sections which intersect in different fashions. Thus, it appeared practical to program the logic needed to model these geometries and set up the input to QUICK. Initially, the nose code was envisioned as a modification to PREKWIK, but the need for a fundamentally different programming approach soon forced the nose geometry problem into a separate computer program. The essential goal of PREKWIK was to receive a minimum of geometric input data via very few input options for a given geometry, perform a few relatively trivial intersection computations, and then generate the card image input required by QUICK. For the programmed geometries, PREKWIK eliminated the tedious chore of preparing a new QUICK deck for every minor quantitative change in geometry (such as a different bluntness, cone half-angle, or vehicle length). In contrast, the nose geometry problem was perceived as logically more complicated. Not only would the nose program have to compute the intersections of more complicated functions, but the capability to round or fillet sharp corners was deemed essential. However, the analytic geometry of rounds and fillets - determing the equation and intersection points of a circle tangent to two intersecting curves - can result in very high degree polynomials along with the attendant logic problems of root sorting to choose the correct intersection. The computational complexity of the intersection problem coupled with the fairly large number of ways a given geometry can be specified via various combinations of known parameters forces the geometry program to accept many different input combinations from which must be computed the same basic set of QUICK input coordinates and slopes. If the nose code did not accept a fairly broad range of input combinations, then the typical user (who didn't have quite the right input combination) might have to spend a great deal of time deriving and debugging the intersection equations. After he had done all this, the computation and assembly of the final QUICK input would seem a fairly minor task. In the meantime, he would have surely forgotten all about the nose code.

Such is the logic upon which is based the structure of the nose geometry code KWIKNOSE, the subject of this user's manual. KWIKNOSE should be viewed as a complement to PREKWIK: whereas the latter is intended to model the overall vehicle, KWIKNOSE is specialized for arbitrary asymmetrical nose shapes. With these two programs, then, a flow program deriving its geometric data from QUICK could handle, as an example, the symmetrically ablated bent cone of Fig. 2. A KWIKNOSE/QUICK/BLUNT/STEIN chain could carry the computation to some point on the forecone, and a PREKWIK/QUICK/STEIN chain could proceed from that point through the bend to the end of the aft cone. (The user may have to construct a fictitious sphere-cone forebody on the PREKWIK model to place the aft body at the same position relative to the geometric orgin of the KWIKNOSE model.)

The asymmetrical nose geometries which KWIKNOSE can generate are limited to cases where the top, bottom, and side body lines can be described in terms of sequential series of straight line and circle segments (see Figs. 3 and 4) as was done for the symmetrical case. The upper and lower half of every cross section must be describable in terms of an ellipse, though the ellipses at successive cross sections need not be concentric or geometrically similar (may have different centers and different major and minor axes). All filleting, rounding, and slicing capabilities are available in the asymmetrical case. However, if the tip of the asymmetric nose does not fall on the centerline of the afterbody (or the first conical section of a stack), the resulting geometry model will be incompatible with the PREKWIK geometry model for the afterbody. However, KWIKNOSE does not distinguish between nose and afterbody sections, and a fairly general asymmetric nose atop a stack of cones and ogives (with bends) can be moldeled entirely with KWIKNOSE without reference to PREKWIK (Fig. 5). Some additional geometric calculation may have to be done by the user.

The remainder of this report describes in greater detail the programmed geometries and the code itself. A description of the input and output is given, and a sample case is presented to illustrate use of the code. The necessary processing information is listed, and certain portions of the programming logic are elaborated upon.

2.0 PROGRAMMED GEOMETRIES

The configurations which can be modeled via KWIKNOSE fall into two general categories: axisymmetric and nonaxisymmetric. The symmetric configuration is an arbitrary sequence of conical and ogival sections which may or may not be tangent at the intersection points. The ogival sections may be convex or concave relative to the centerline of the vehicle. Nontangent intersections may, at the user's option, be smoothed with circular fillets or rounds. The fillets (or rounds) may be inserted between cone-cone intersections, cone-ogive intersections, or ogive-ogive intersections. The fillets need not be very small in relation to the intersecting curves and may therefore form major segments of the body. The only restriction is that the adjacent segments being filleted intersect in the real plane. For some cone-ogive and ogive-ogive combinations, there may exist a maximum filleting radius beyond which a real solution does not exist. In general, no segment may contain a point with infinite slope, the nosetip being the one exception. The tip of the nose may be either sharp or blunt and is always used as the geometric origin for the entire geometry. No segment may wrap back on itself so as to require a double-valued radius from any single point on the centerline axis. The program dimensions limit the geometry model to ten or fewer segments, and each fillet occupies one segment (see Table 1, note 8). Since KWIKNOSE does not distinguish between nose and afterbody portions of the vehicle, the user may allocate the ten segments between these two portions as needed.

The program contains logic to insert slicing planes into the basic axisymmetric vehicle described above. Provisions are made for an upper slice, a side slice, and a lower slice, each being composed of nine or fewer straight-line segments. In the side view the top and bottom slicing planes are seen as segmented lines, and the side slicing plane is seen as a segmented line in the top view. Thus, there are no provisions for skewed slices. The slices may not cross the centerline of the vehicle and may not intersect each other inside the region of the unsliced vehicle at a particular station. Corners at the intersection of adjacent segments may be rounded or filleted at the user's option. Although the slices are assumed to be planar, the fillets need not be small and may therefore comprise major portions of the slicing segments. It should be understood that the slicing body lines are input to QUICK independently of the other body lines and thus do not have to intersect the unsliced vehicle at all points or any points. QUICK automatically checks whether the slicing body lines intersect the unsliced body and chooses the appropriate curve for the outer skin of the vehicle. The slices therefore may enter and leave the body as needed. There are no provisions for filleting or rounding in the circumferential direction (as viewed in planes normal to the centerline of the vehicle).

The asymmetrical case retains all combinations of cone and ogive intersection options (or line-circle intersections in a particular radial plane), the rounding and filleting of the outer body lines, and slicing with rounds and fillets. The asymmetrical vehicles treatable by KWIKNOSE must be such that the top, bottom, and side surface body lines may be represented by segments of lines and circles. When viewed in cross section, the vehicle must be describable with two ellipses (not necessarily identical) for the upper and lower half-planes. At a given cross section, the horizontal line dividing the two half-planes need not intersect the centerline (or geometric axis through the nosetip), and provisions are available for defining this deviation. For the asymmetrical case the top and bottom surface body lines may cross the geometric axis, and the map axis (from which QUICK swings the radius vector which must remain single valued) may also depart from the geometric axis if needed. Thus, the asymmetrical option allows, for example, the modeling of an ablated nose atop a bent cone.

For certain asymmetric configurations there is a potential inaccuracy of which the user should be aware. Consider a geometry such as is illustrated in Fig. 3 but without the round between the fore and aft sections. This geometric model was constructed with the geometric axis (X-axis) through the nosetip and parallel to (but not coincident with) the aft cone centerline. In the actual hardware (before rounding), the intersection of the fore and aft sections when viewed from the side would appear as a slanted line. Discontinuities in the surface slope would occur at this line in moving along a body line. In the math model, however, the slope discontinuity would occur only on the upper, lower, and side body lines and would be smeared everywhere in between. This smearing results from two conditions: (1) QUICK assembles cross sections in planes normal to the geometric axis; (2) as modeled in Fig. 3, the slanted line between the fore and aft bodies is oblique to the geometric axis. While a carefully constructed manual model for input to QUICK probably could represent this case accurately, programming of the general case for arbitrary intersections of cones and ogives was considered beyond the scope of KWIKNOSE.

The user has one alternative if he requires an accurate model from KWIKNOSE: provide the input data relative to a geometric axis normal to the slanted line. Note, however, that the geometric axis might penetrate the skin and require bending of the map axis to keep QUICK's radius vector single-valued. In addition, once the geometric axis has been made normal to a particular intersection plane, no other similar intersection with a different slant angle can be modeled with equal accuracy. For two conical sections of different half-angles and oblique centerlines, the intersection "plane" is really curved. Thus, reorienting the geomtric axis can only reduce (but not eliminate) the inaccuracy in modeling.

3.0 PROGRAM DESCRIPTION

3.1 INPUT

The input to KWIKNOSE is described in Tables 1 through 4, and a listing of the KWIKNOSE/QUICK jobstream with the input for the sample case of Figs. 4 through 6 is given in Table 5. It is suggested that the user, prior to assembling the input deck, make a qualitatively accurate sketch of the upper, lower, and side body lines and indicate the known geometric data. This task has been found useful in choosing appropriate input options and deciding on the values of the logical input variables (DIR and SLOPE), which are needed by the code for selection of a segment intersection from multiple roots.

The first input card is for identification and may contain any 80 characters, though QUICK will carry only 60 of them. The remaining input is via namelist /INPUT/. For a symmetrical body (with or without asymmetrical slices), only one namelist input is required. For an asymmetrical body three separate namelists must follow the single identification card. The first must define the upper body line (ZUCL, see Figs. 7 and 8), the second the lower body line (ZLCL), and the last the side body line (YSID). Note that the input variable Z is used for the radial coordinate regardless of whether the input data represents the Y or Z direction. The slicing body lines (ZUSL, ZLSL, and YSSL) and the Z coordinates of the side and map axis (ZSID and ZMAPAX) may be included in any of the three namelists, if they are input at all. The upper, lower, and side body lines are processed independently of each other and need have no particular correspondence with each other except for the same final X coordinate at the end of the last segment. Thus, each may be an entirely different combination of lines, circles, and fillets. The sign notation for input coordinates and angles is that of a cylindrical coordinate system. Angles (TH(I)) are positive away from the centerline and negative toward it relative to a line parallel to the centerline.

The input rules for the three main body lines are somewhat different than for the slicing body lines and will be discussed separately. For each of the three main body lines (upper, lower, side) to be defined, the user must input the element of the character array CURVE(I) as 'LINE' or 'OGIV' for each segment. Based on this choice for each individual segment, the user then chooses an input option from Table 2 for each of the segments, preferably an option which matches the combination of geometric data he has for the segment. If any redundant data is to be input, case 17 or 18 (see ICASE values in Table 2) must be chosen. If the redundant data is not self consistent to within ERRMAX, execution will be halted. All data items indicated by X's in Table 2 for a particular ICASE value must be input or execution will be halted. Note that ICASE(I) is not an input variable. The data elements described in Table 1 are then coded according to the standard rules for namelist input as found in any Fortran manual. The user should note that X(I), Z(I), and TH(I) are the coordinates and slope at the end of the Ith segment (CURVE(I)). Further, certain input cases may not be used following certain other cases. For example, cases 5 and 9 constitute inputting only the equation of the segment and in themselves do not include sufficient information to locate the intersection of the end of the segment. Thus, those cases which assume that the coordinates of the end of the previous segment are known may not follow cases 5 and 9. Such limitations, the need for which will be clarified in Section 3.4, are given in the footnotes to Tables 1 and 2.

The direction indicator variable DIR(I) requires additional clarification. Whenever ogival segments are involved, the possibility of multiple intersections with adjacent segments exists. In addition, when an ogive is to be inserted between two given points, there are usually two circle equations which will satisfy all the numeric data. The user therefore must indicate to KWIKNOSE which of the possibilities is to be chosen using DIR(I). Table 2 indicates which input cases require DIR(I) and Table 3 defines the acceptable values of DIR(I) for each ICASE value. The nomenclature for these acceptable values is defined in Table 4. The 'OPUP', 'OPDN', 'TOPS', and 'BOTS' values apply to all three main body lines (upper, lower, and side) as though they were rotated about the centerline for viewing as oriented in Table 4. That is, DIR(I) = 'OPUP' indicates that the centerline sees a convex segment and 'OPDN' a concave segment. Similarly, 'TOPS' always indicates the intersection farther from the centerline than the 'BOTS' intersection.

If a nontangent intersection is to be filleted or rounded, this fact is indicated by giving the radius of the round or fillet RHO(I) to be inserted at the end of the Ith segment. Fillets/rounds are not included when numbering segments for input, and no data other than the radius are input for the fillet. The intersections and center are computed internally by KWIKNOSE. There are several restrictions on the filleting (or rounding) process. Segments already tangent may not be filleted. Only adjacent segments which actually intersect may be filleted because the location of this intersection is used to choose between multiple roots. Further, some types of intersections may have maximum filleting radii beyond which real solutions do not exist. If KWIKNOSE detects any of these conditions, usually through negative square roots or division by zero, execution will be halted.

For asymmetrical bodies, such as a bent cone, the user may opt to define the Z-coordinate of the side (ZSID, Figs. 6 and 8) as different from zero, in which case he must input two or more values of XSID and ZSID as described in Table 1. XSID(1) and ZSID(1) must both be zero; and if ZSID is to be input at all, at least two values of each (XSID(1) and XSID(2), ZSID(1) and ZSID(2) must be given. If the vehicle is bent so far that the geometric axis penetrates the skin, the user must bend the map axis via XMAPAX and ZMAPAX so that a radius vector swung about the map axis at any station along the vehicle will be single-valued. Care should be taken in using models generated by QUICK in which the map axis is bent. The stock version of the QUICKLOK phase of QUICK returns to the user program radius vectors relative to the map axis rather than the geometric axis. Thus, a bent cone whose map axis follows the centerlines of the cones may appear to the user program as not being bent at all. Input follows the same rules as for XSID and ZSID above. Note that both ZSID and ZMAPAX may be negative. These variables are used only if TYPE = 'ASYM', for which case ZSID defaults to zero if not input, and ZMAPAX defaults to ZSID if not input. These input variables may be included in any one of the three namelists.

Input for the slicing planes as well as the processing of the data by KWIKNOSE is completely independent of the main body lines. Any combination of the three slices may be used regardless of the symmetry aside from the slices (the value of TYPE has nothing to do with symmetry of the slices). The minimum slicing plane input, if present at all, is the coordinates of the beginning and end of one segment. At the user's option, different combinations of X and Z (or Y) coordinates and slope angle may be input. Note that TUSL(I) is the slope angle of the segment ending at XUSL(I), ZUSL(I), for I > 1. Thus, TUSL(1) = TUSL(2), and similarly for the other two slices. Redundant data will be checked for consistency; but if any one of the three is not input for a given segment, it will be computed. The slicing planes need not cover the entire length of the vehicle, but the first and last segments will be extrapolated to the extremities if not. Nontangent intersections of adjacent segments may be filleted or rounded by inputting the fillet/round radius as indicated in Table 1. Note that RHOUS(I) fillets the intersection at XUSL(I), which is the end of the I-1st segment; otherwise, all the rules of body line filleting/rounding apply.

The remaining input variables are program control integers. IPRINT set different than zero will cause KWIKNOSE to add cards to the QUICK input deck to exercise the math model and generate printouts of coordinate and slope data at various points along each body line segment. IPRINT=0 (the default) suppresses such printout. The variable IR is the unit number (Data Set Reference Number) of the facility card reader, and IW is the unit number of the printer. IF is the unit number of the KWIKNOSE output file of card images to be passed to and read by the QUICKDEF phase of QUICK. IP is the unit number for the facility card punch. If IF and IP are different, the data from unit IR is copied as card images with an 80-character array to IF, from which the input cards are then printed on IW. IF is then rewound and read as namelist data.

3.2 OUTPUT

Table 6 is the KWIKNOSE printout for the sample case. The printout is composed of three sections: the input data, the results of the computations, and the QUICK input file. The first section, labeled "INPUT CARDS" is a listing of the data cards exactly as punched. The data between the first two lines of periods is the input data after the first namelist has been read. Notice that geometric variables not input through the namelist default to 1.E70. This defaulting is critical to the program's logic and should not be trampered with by inputting user-chosen default values which have no geometric meaning. The printout from this point to the third line of periods contains the results of the geometric computations. First, the number of segments exclusive of rounds and fillets is printed followed by the code's determination as to the input options the user has chosen for each segment. Next, the results of the computation of segment intersections and equations are printed. In the printout labeled "SEGMENT END POINT COORDINATES AND SLOPES", X and Z are the coordinates at the beginning of the segment, except for the last point (with values less than 1.E70), which is the end of the last segment. TA and TF are the slope angles in degrees at the aft and fore ends, respectively, of the segment. The segment equations in the next section of the printout are defined in terms of the center coordinates and radius for ogival sections, and slope angle and Z-intercept for straight-line segments. The X value printed here is the coordinate of the end of the segment. In these terms the line and ogive equations are

$$\text{line:}\quad Z = X \tan T + B$$

$$\text{ogive:}\quad (X-H)^2 + (Z-K)^2 = R^2$$

Next, the results of the fillet and round computation is printed. The equation of the fillet is defined as

$$\text{fillet:}\quad (X-KSI)^2 + (Z-ETA)^2 = RHO^2$$

and straddles the printed X value. X1, Z1, T1, and X2, Z2, T2 are the coordinates and slope angle (degrees) at the fore and aft intersections, respectively, that is, the points of tangency with the intersected segments. The fillet/round intersections are then printed as additional segments included with the previous unfilleted body line under the heading "FINAL QUICK ARRAY". Note that the original intersection point has been removed from the arrays. This completes the geometric computation for the upper body line (ZUCL). When TYPE = 'ASYM' in the namelist input, this computation and printout are repeated first for the lower body line (ZLCL) and then for the side (YSID). The third printout includes the results of the geometrical calculation for slices. The section of the printout headed "SLICE GEOMETRY" gives the slope angle for each segment ending at the indicated coordinate values. These results are exclusive of rounds or fillets, for which immediately follows the computational results. BI and BJ are the Z-intercepts of the two lines formed by the locus of the fillet/round center while rolling on the two intersecting segments. A3 and A4 are the Z and X coordinates of the fillet/round center; and X1, Z1, T1, etc., are the tangency points as described for the main body line fillet/round printout. The final portion of the computational printout is the slicing body lines with the rounds and fillets inserted. The KWIKNOSE printout concludes with a listing of the card images of the QUICK input file. This listing will not be generated if IF = IP. The contents of this output deck may be understood from Table 7 and Refs. 1 through 3 and will not be discussed here. The user should understand that this deck comprises the input to the QUICKDEF phase of QUICK and not the QUICKLOK phase, which is the portion of QUICK employed in the fluid dynamics programs to obtain geometric information.

3.3 PROCESSING INFORMATION

KWIKNOSE is a Fortran IV program containing about 2,000 cards (Table 8). The code was developed on an IBM 370/165 computer system under the H Level 21.7 compiler. Compilation of the entire deck requires 230 K bytes of core and 40 sec of central processing unit (CPU) time. Execution of the code requires 80 K bytes of core, and all geometries processed to date have required less than 2 sec of CPU time. Users should note that because of the character manipulation performed in the program, it cannot be compiled under the G level 21 compiler. Execution requires one peripheral storage device for unit IF (defaulted to unit 1). This should be a sequential file formatted for 80-character records as illustrated in Table 5.

3.4 PROGRAMMING CONSIDERATIONS

The overall flow of logic in KWIKNOSE is basically simple and can be best understood via the series of CALL statements in the main program (Table 8) and the subroutine description of Table 9. Execution begins with the initialization of certain data (subroutine INIT), after which the first namelist is read (INPT). Next, the input option chosen by the user is determined for each segment (CASE) in terms of the value of ICASE(I). This variable is then used to control branching in determining the segment equations (SEGE) and locating the segment intersections (INTR). This completed, the results of the computation to this point are copied into the QUICK arrays XQ, ZQ, and TQ (QARY) and printed (PRNT). Next, the round/fillet computations are performed and added to the QUICK arrays (ROFL). If slices are present (indicated by ISL ≠ 0), the necessary computations are performed and the results assembled in the QUICK slicing array V (SLCE). If rounds or fillets are present on the slices, these computations are performed and the results inserted into the original arrays (RFSL). If the case is symmetrical, the QUICK input card image file is generated (QUCK) and printed (LIST). If the case is asymmetrical, the completed body line with rounds and fillets is copied into the three-dimensional QUICK array VB (COPY). When all three main body lines are complete, the slicing with optional rounds/fillets is performed and copied into the three-dimensional arrays. Next, ZSID and

ZMAPAX are defaulted if not input, and the remaining body lines are checked for validity (BLCK). Finally, the QUICK input file for the asymmetrical case is generated (QK3D) and listed (LIST), thus completing the processing.

For a more detailed examination of the code, the variable description in Table 10 may be useful. Users interested in the intersection computation should find the logic as coded in SEGE and INTR fairly straightforward, but a few clarifying comments about the rounding and filleting logic are necessary. The straightforward approach to locating the center and tangency points of a circle filleting two other circles is to use analytic geometry, by writing four circle equations and two tangency equations in terms of six unknowns. However, this is equivalent to a polynomial of degree 48 for one unknown, which is an absurd thing to try to solve. The approach used in ROFL is to write the equation of the locus of fillet centers as the fillet rolls on the two adjacent segments, where the intersection of these two loci of centers is the center of the filleting circle. This approach reduces the problem to second degree but requires additional logic to decide when to add or subtract the fillet radius to or from the segment radius (ogive) or Z-intercept (line). Fortunately, the information necessary to make this decision can be extracted from the slopes of the adjacent segments at the intersection point. For the line-ogive intersection, the final second degree problem can then be solved by choosing the set of tangency points which straddle the intersection point. For the ogive-ogive intersection, the filleting circle is chosen whose center is closest to the original intersection point. The advantages of using the loci-of-centers approach are clear; but the major disadvantage is that segments which do not intersect cannot be filleted, whereas the general 48th degree problem includes this case.

Because of the complexity of the above logic as coded and to the many input options available, KWIKNOSE contains many small blocks of coding to check on the validity of the input and the running computation. When a violation of the rules is detected, a condition code is generated with the Fortran statement STOP xxxx, where xxxx is the four-digit condition code. Execution is halted and an error message is usually generated. A program-generated condition code without an accompanying error message would indicate either a programming error or computer error (unlikely). The condition code is usually the same as the statement number of the format that generated the accompanying message. For five-digit statement numbers, the left most digit is truncated to obtain the condition code.

The accuracy of the code's geometry computation has been checked by comparison with manual computation on an electronic calculator. To the best of the author's knowledge, each block of coding has been checked against a tailored check case. However, the user is encouraged to carefully scrutinize the final QUICK math model for accuracy. The author has, to date, constructed about 30 manual card input decks to QUICK and hundreds of PREKWIK and KWIKNOSE decks, and no errors have been traced to QUICK. Thus, errors in the final math model will most likely be caused by either problems in the setup codes or their input decks. Checking of the QUICK math model can be effectively accomplished if a plotting device is available with stock software. With very little effort, the author has constructed a cathode ray tube (CRT) plotting program using the QUICKLOK portion of QUICK and has found it indispensable in debugging the geometry models. This approach is highly recommended to all users.

4.0 SUMMARY

A user's manual has been presented for a Fortran computer program KWIKNOSE which, for certain axisymmetric and asymmetric nose configurations, provides simplified geometric input to the Grumman QUICK-geometry code, which in turn provides a geometric math model to various numerical flow-field programs. A discussion was presented as to how KWIKNOSE may easily be used to model any vehicle whose upper, lower, and side body lines can be described in terms of sequential segments of circles and straight lines and whose upper and lower half cross sections can be modeled with ellipses. It was further shown how the body may be sliced on three sides. A description was given of the input and output, and a sample case was presented to illustrate use of the code.

REFERENCES

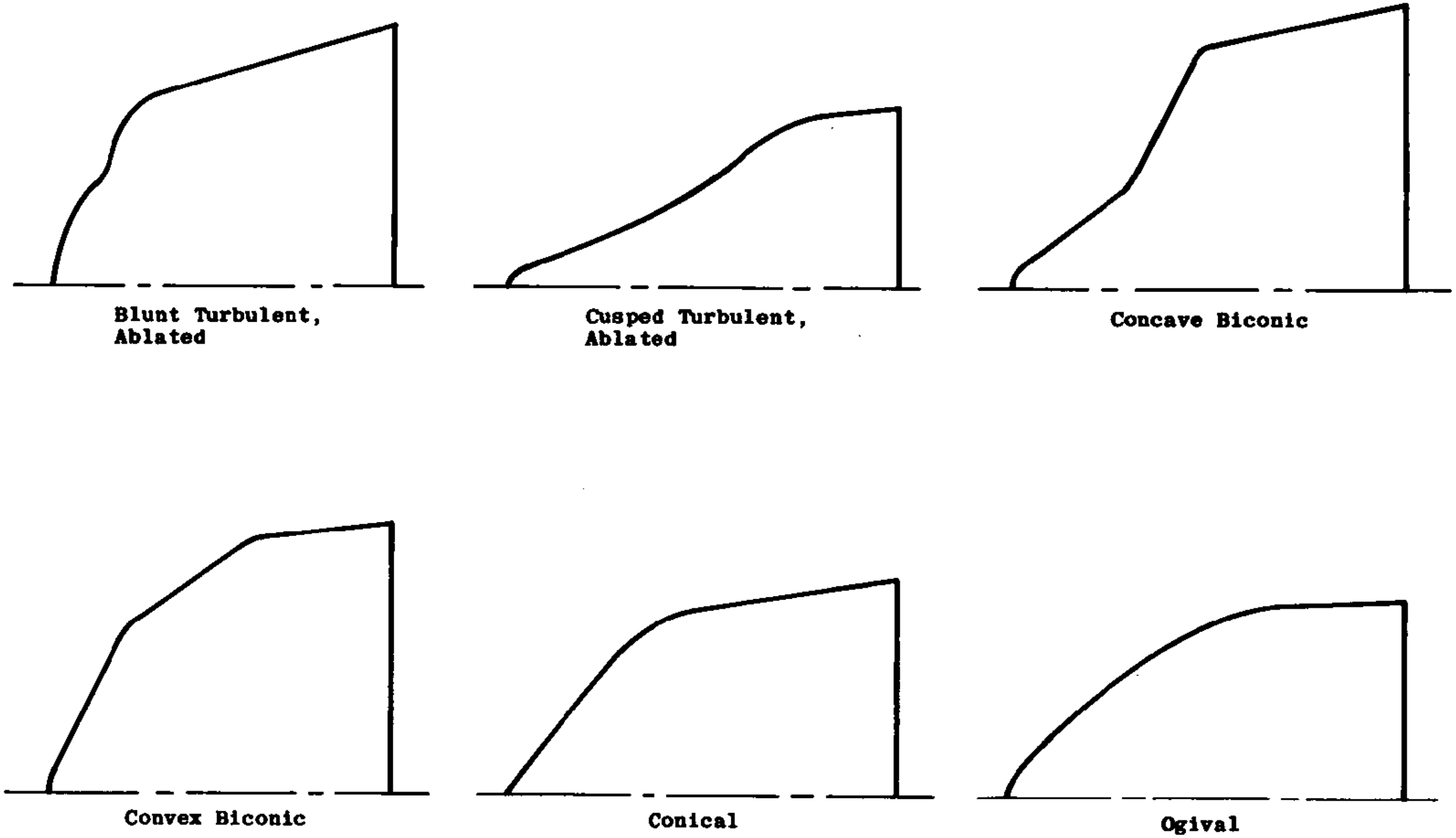
1. Vachris, A. and Yaeger, L. "QUICK-GEOMETRY User's Manual." Grumman Aerospace/Aerodynamic Section Technical Data Report No. 393-74-1, 1974.

2. Vachris, A. F. and Yaeger, L. S. "QUICK-GEOMETRY: A Rapid Response Method for Mathematically Modeling Configuration Geometry." NASA SP-390, 1975, pp. 49-73.

3. Marconi, F. and Yaeger, L. "Development of a Computer Code for Calculating the Steady Super/Hypersonic Inviscid Flow Around Real Configurations, Volume II - Code Description." NASA CR-2676, May 1976.

4. Marconi, F., Salas, M., and Yaeger, L. "Development of a Computer Code for Calculating the Steady Super/Hypersonic Inviscid Flow Around Real Configurations, Volume I - Computational Techniques." NASA CR-2675, April 1976.

5. Kutler, P., Reinhardt, W. A., and Warming, R. F. "Numerical Computation of Multishocked, Three-Dimensional, Supersonic Flow Fields with Real Gas Effects." AIAA Paper No. 72-702, June 1972.



Blunt Turbulent, Ablated

Cusped Turbulent, Ablated

Concave Biconic

Convex Biconic

Conical

Ogival

Figure 1. Typical geometries available via KWIKNOSE.

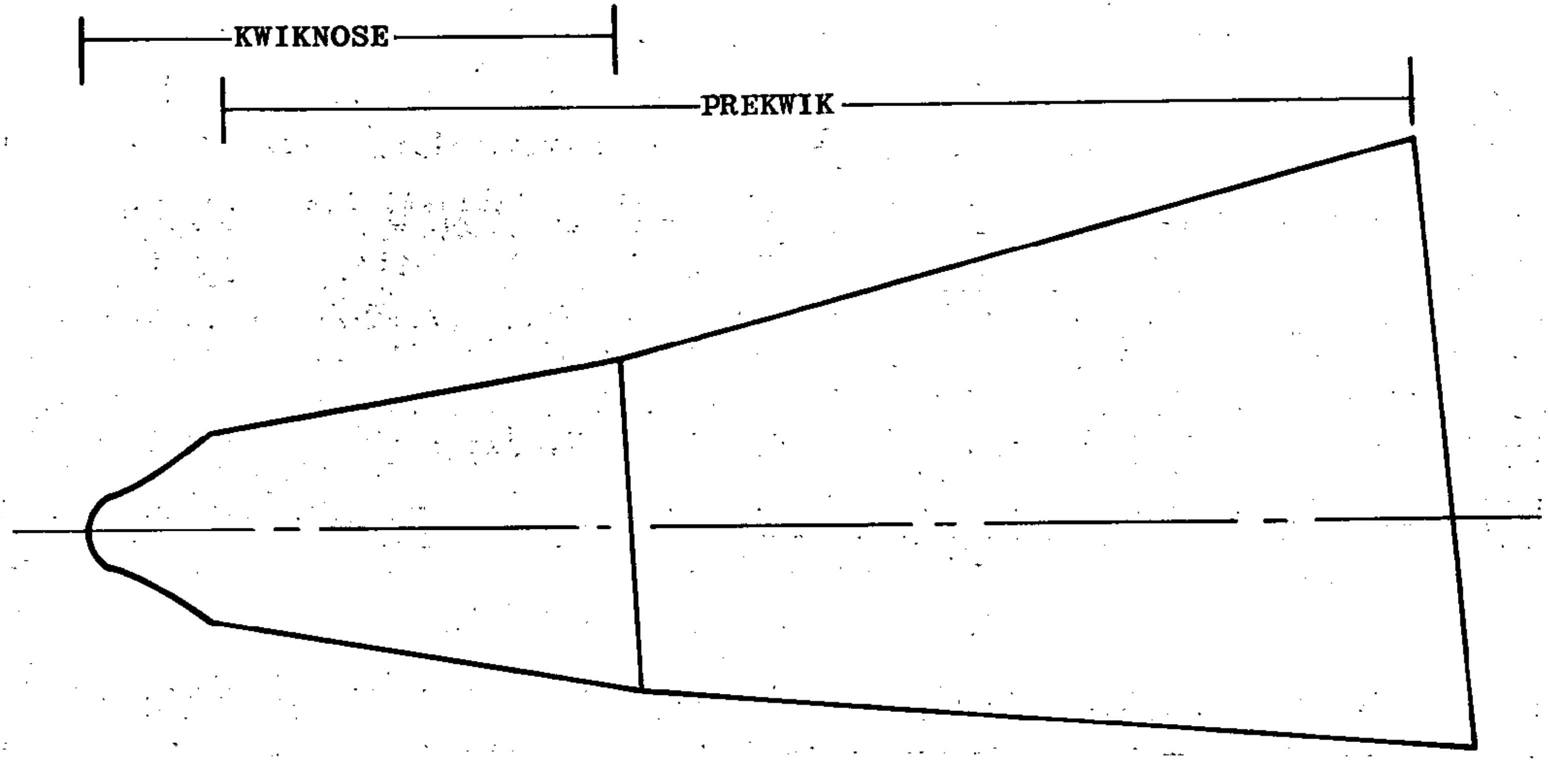


Figure 2. Relative regions of applicability of programs KWIKNOSE and PREKWIK.

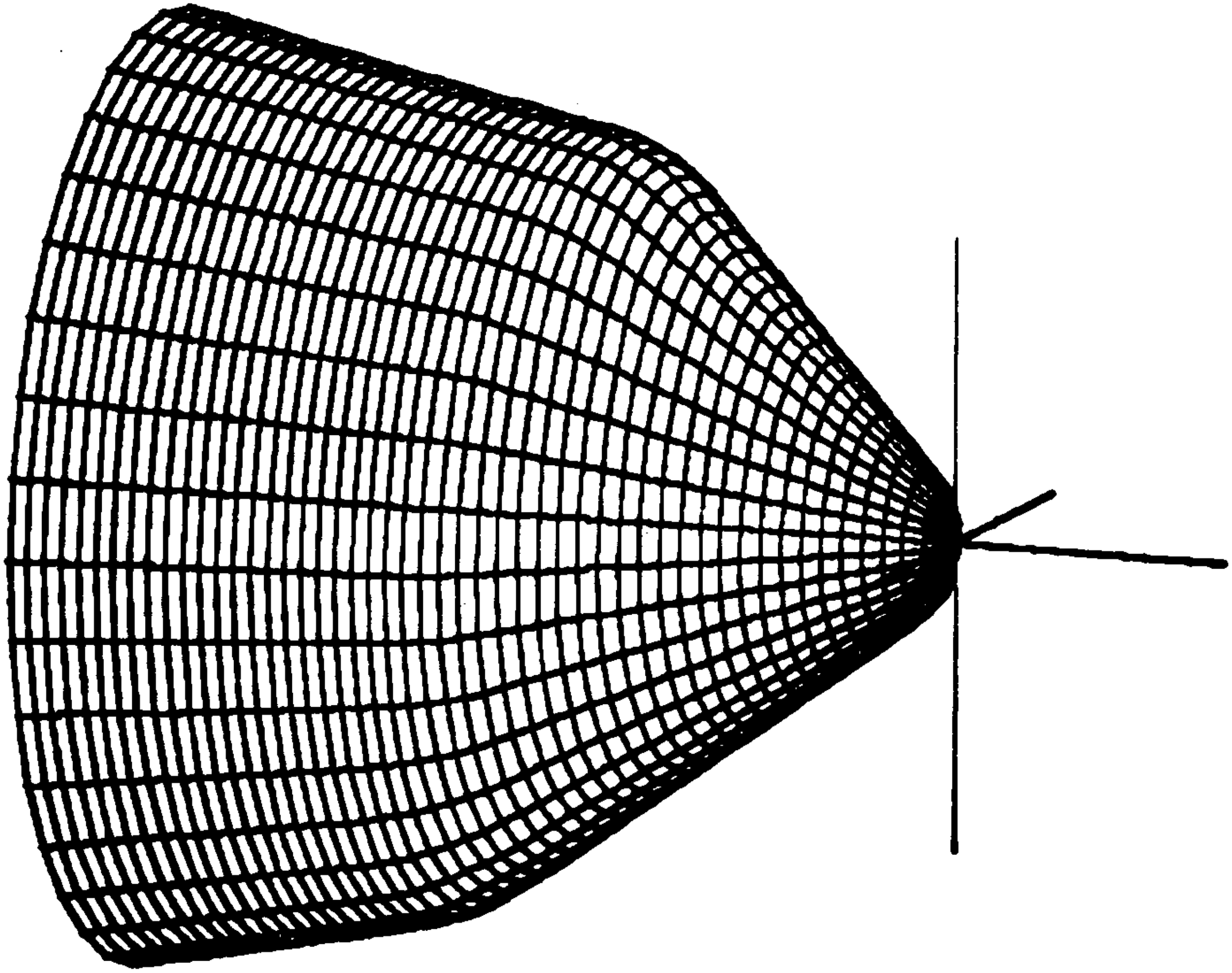


Figure 3. Oblique view of asymmetrically ablated nose.

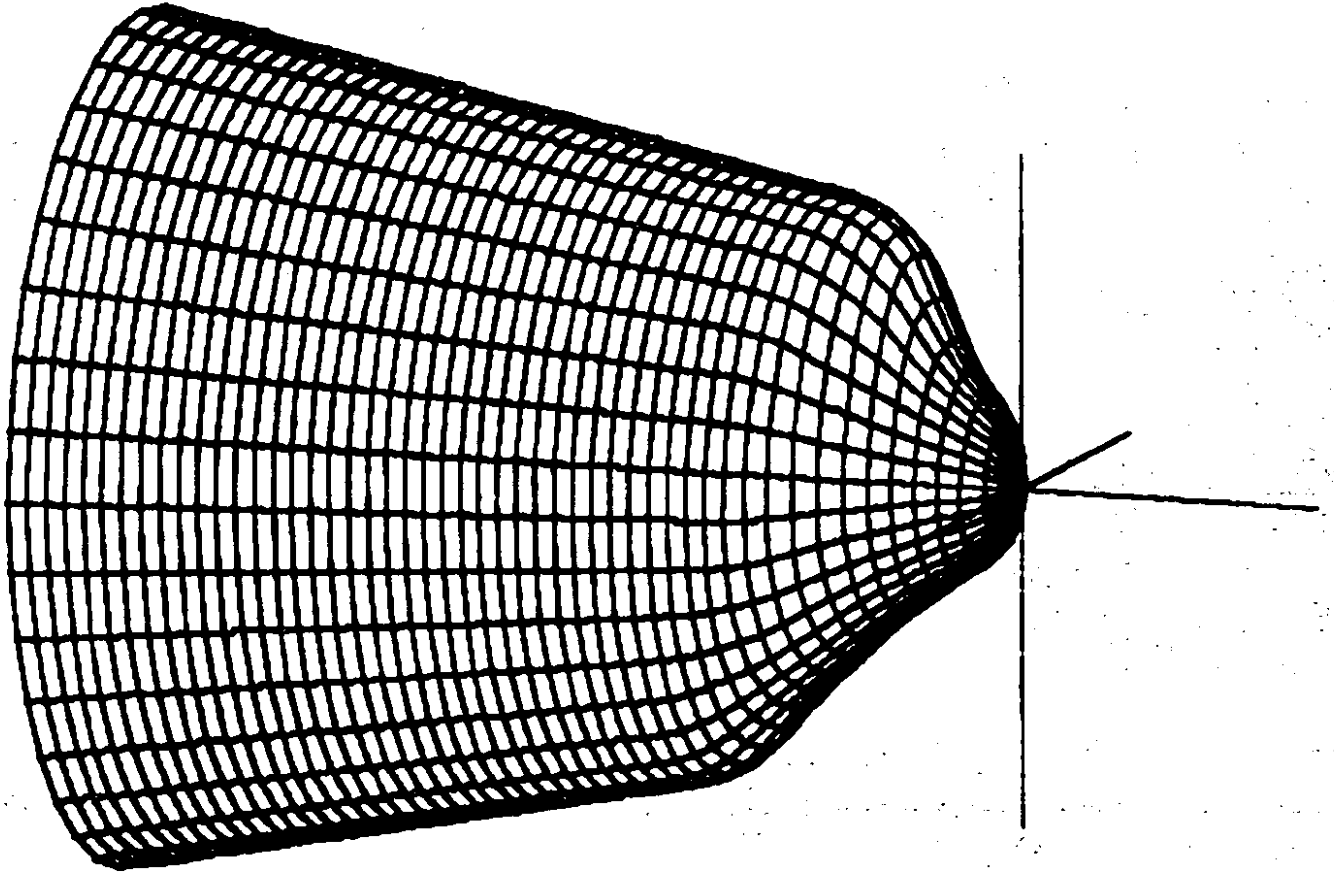


Figure 4. Oblique view of asymmetrical nose from sample case.

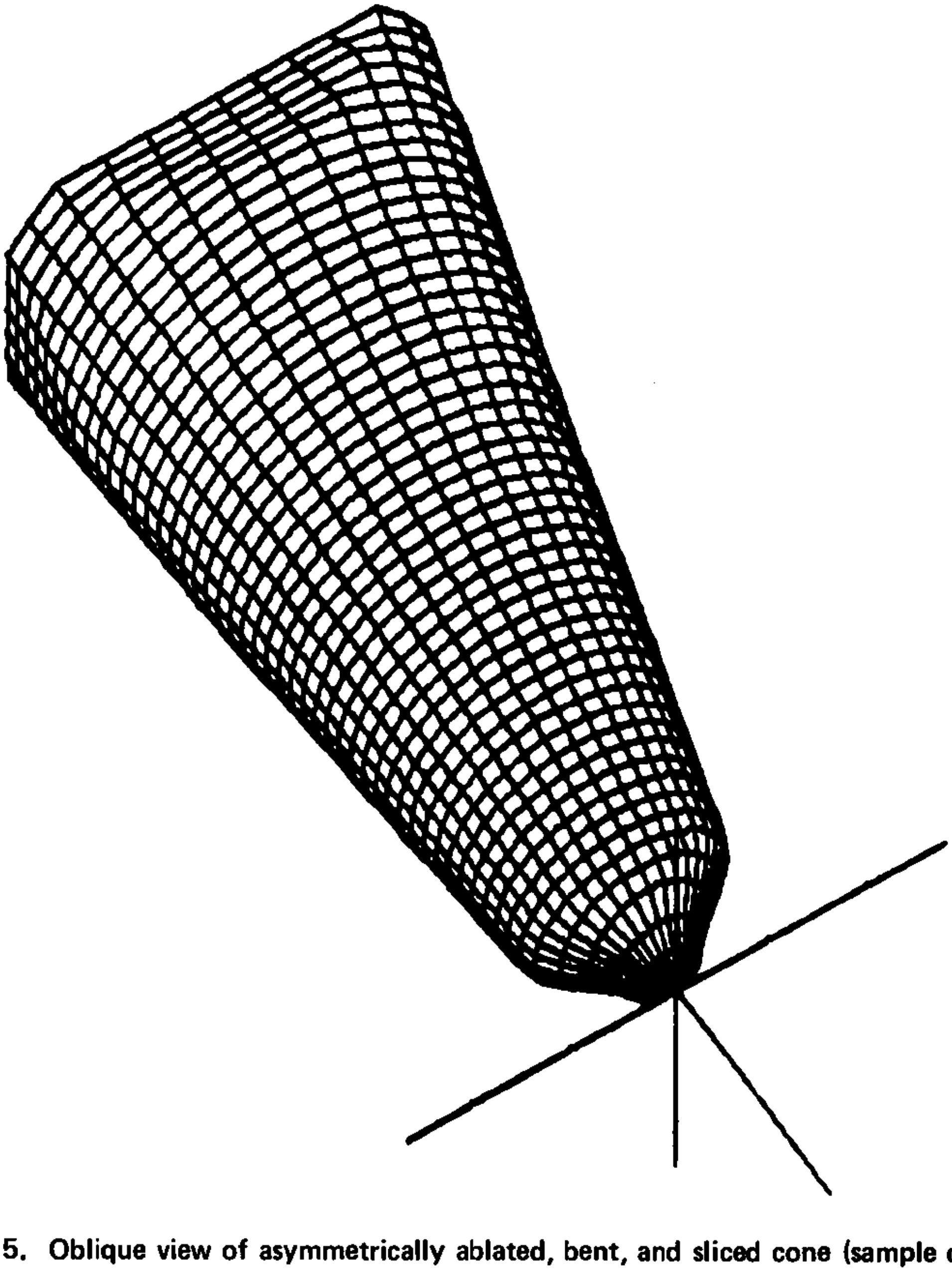


Figure 5. Oblique view of asymmetrically ablated, bent, and sliced cone (sample case).

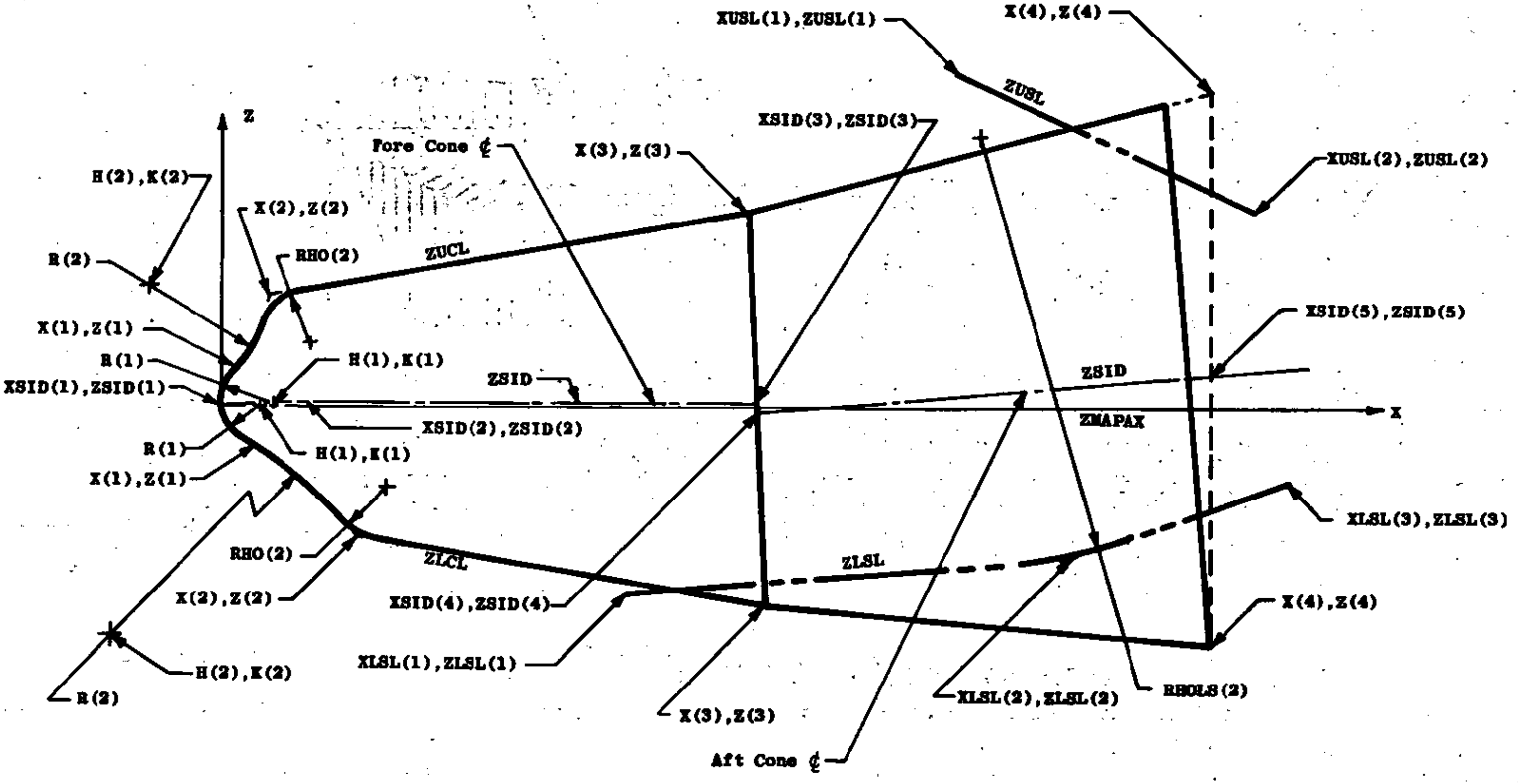
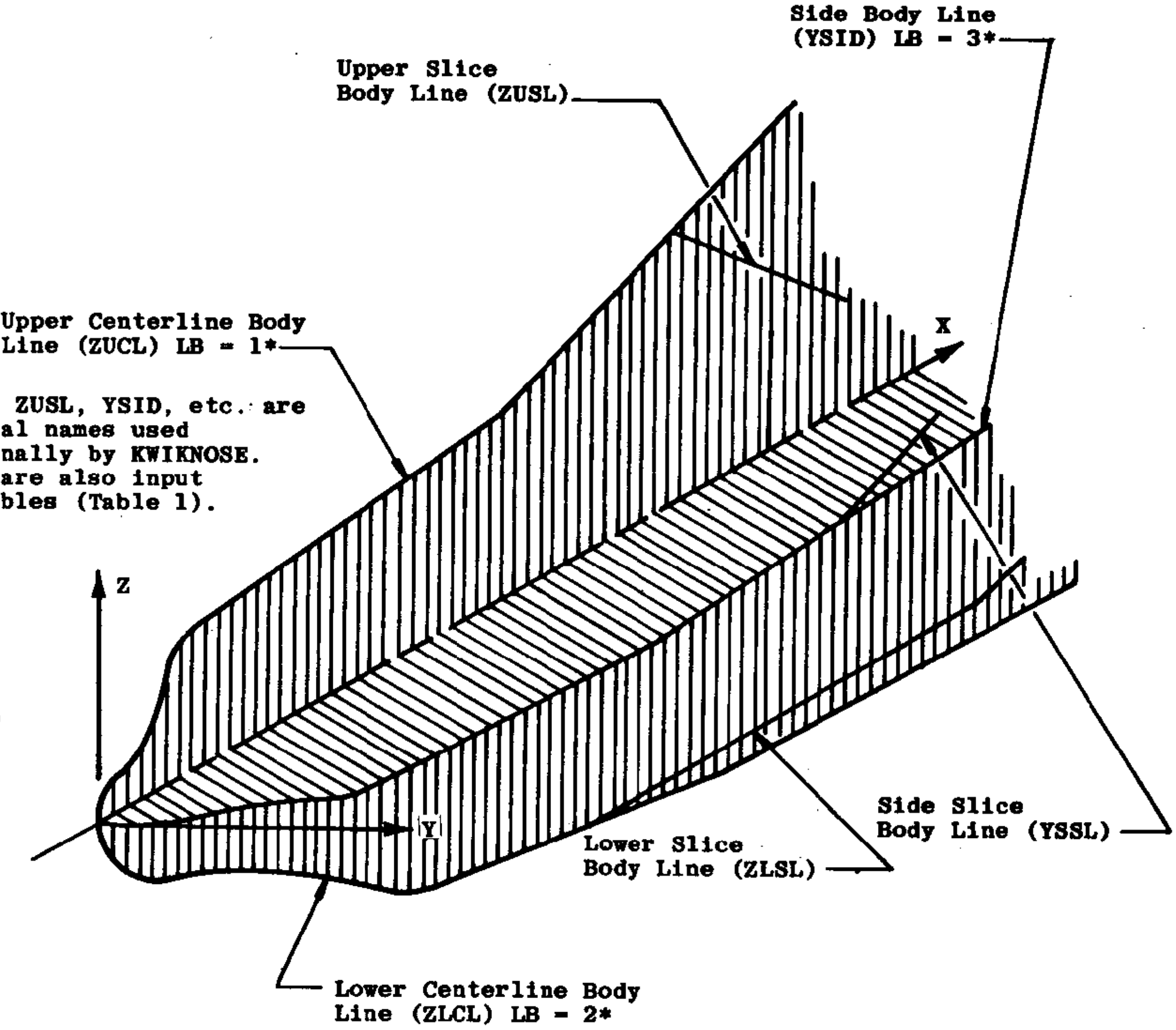


Figure 6. Nose-afterbody sample case with notation for input variables.



Note: ZUCL, ZUSL, YSID, etc. are logical names used internally by KWIKNOSE. Some are also input variables (Table 1).

*See Table 10

Figure 7. Body line notation.

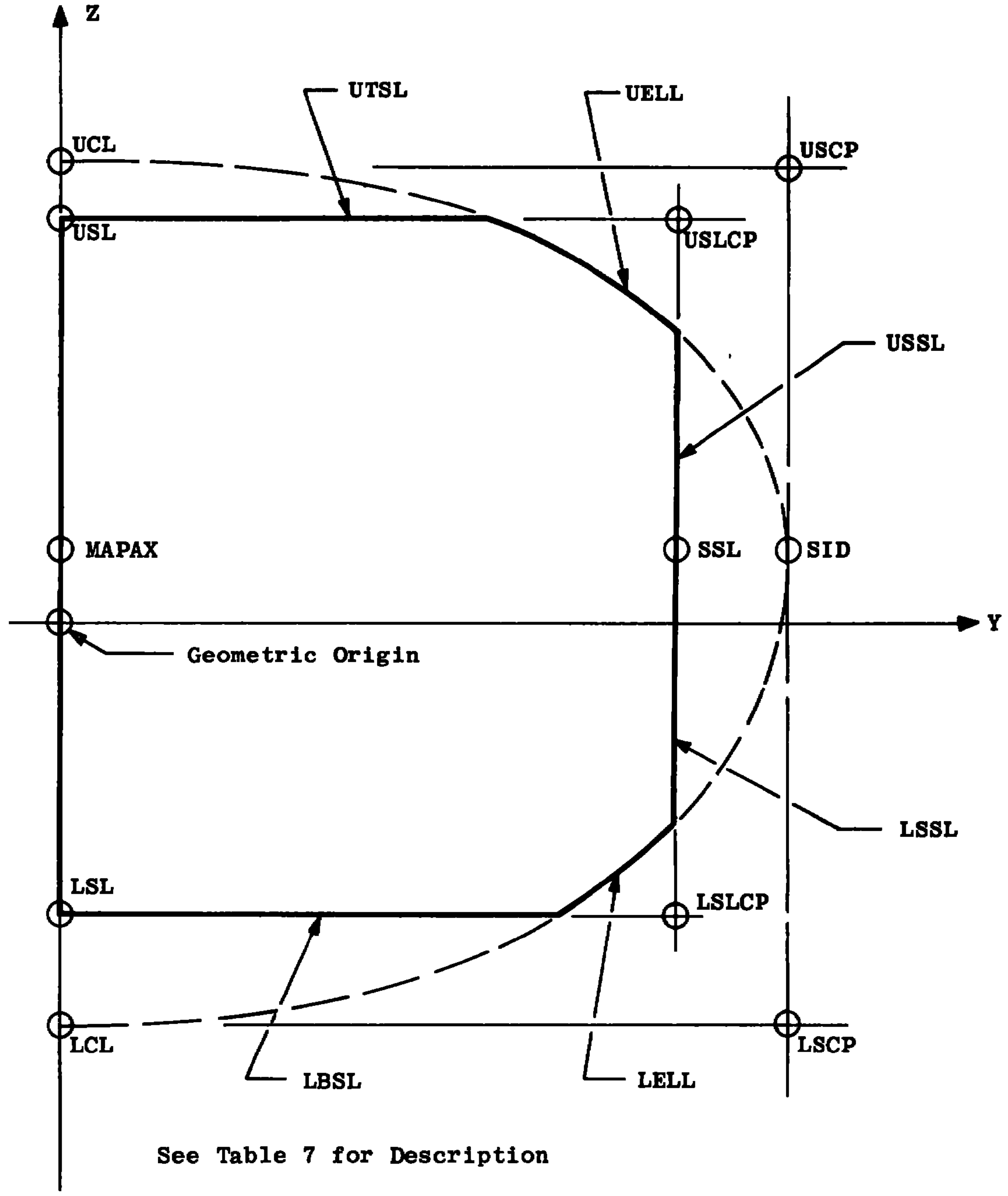


Figure 8. Cross section and body line models.

Table 1. Input to KWIKNOSE via Namelist/INPUT/

Namelist Variable	Value	Default Value	Meaning	TYPE[7]
X(I)	(0,∞)	1.E70	Centerline distance to end of Ith segment (I≤10)	R*4
Z(I)	[0,∞)[9]	1.E70	Radial distance (normal to centerline) to end of Ith segment	R*4
TH(I)[2]	(-90,90)	1.E70	Slope angle (deg) at end of Ith segment, positive away from centerline	R*4
CURVE(I)	'LINE'	'bbbb'[1]	Segment I is a straight line	R*4
	'ØGIV'		Segment I is an ogive	
R(I)[3]	(0,∞)	1.E70	Radius of ogive of Ith segment (CURVE(I)='ØGIV')	R*4
H(I)[3]	(-∞,∞)	1.E70	X coordinate of center of ogive (CURVE(I)='ØGIV')	R*4
K(I)[3]	(-∞,∞)	1.E70	Z coordinate of center of ogive (CURVE(I)='ØGIV')	R*4
B(I)[2]	(-∞,∞)	1.E70	Z intercept of line segment (CURVE='LINE')	R*4
DIR(I)[4]	See Table 3	'ØPDN'	Direction indicator for CURVE(I)='ØGIV' when aft intersection with adjacent segment may have two solutions	R*4
SLØPE(I)	'T'	'F'	Indicates ogive is to be made tangent to aft end of previous segment (do not use for segment I=1)	L*1
ERRMAX	$>10^{-7}$	5.E-5	Error criterion for checking consistency of redundant input data	R*4
RHØ(I)[5,8]	(0,∞)	1.E70	Fillet/round radius at end of Ith segment	R*4
XUSL(I)[5]	[0,∞)	1.E70	X and Z coordinates and slope (deg) of upper slicing plane, I≤10	R*4
ZUSL(I)[5]	[0,∞)	1.E70		R*4
TUSL(I)[5]	(-90°,+90°)	1.E70		R*4
XLSL(I)[5]	[0,∞)	1.E70	X and Z coordinates and slope (deg) of lower slicing plane, I≤10	R*4
ZLSL(I)[5]	[0,∞)	1.E70		R*4
TLSL(I)[5]	(-90°,+90°)	1.E70		R*4
XSSL(I)[5]	0,∞)	1.E70	X and Y coordinates and slope (deg) of side slicing plane, I≤10	R*4
YSSL(I)[5]	[0,∞)	1.E70		R*4
TSSL(I)[5]	(-90°,90°)	1.E70		R*4

Table 1. Continued

Namelist Variable	Value	Default Value	Meaning	TYPE[7]
RHØUS(I)[5]	(0,∞)	1.E70	Fillet/round radius on the upper, lower, and side slices,	R*4
RHØLS(I)[5]	(0,∞)	1.E70	respectively. RHØ—(I) fillets or rounds the <u>beginning</u> of	R*4
RHØSS(I)[5]	(0,∞)	1.E70	segment I and the end of segment I-1, 2<I<10.	R*4
XSID(I)[6]	[0,∞)		X and Z coordinates of side surface body line. Input only if body is asymmetrical and side surface	R*4
ZSID(I)	(−∞,∞)		body line is not in some horizontal plane as the geometric X axis. Defaults to X axis (see Fig. 8).	R*4
XMAPAX(I)	[0,∞)		X and Z coordinates of QUICK map axis (the axis relative to which the vehicle must be describable in	R*4
ZMAPAX(I)	(−∞,∞)		terms of a single-valued radius vector). Input only if body is asymmetrical and single-valued radius	R*4
			requirement cannot be met by default. Default to XSID, ZSID.	
TYPE	'SYMM'	'SYMM'	Vehicle is axially symmetric. Provide input for one body line.	R*4
	'ASYM'		Vehicle is asymmetric. Provide input for three body lines: UCL, LCL, and SID (see Fig. 8). Input the needed combinations of CURVE, X, Z, TH, R, H, K, B, DIR, and SLØPE in three successive &INPUT namelists. Provide slicing data, XSID, ZSID, XMAPAX, ZMAPAX with any one of three if required. First of three namelists is upper centerline, second is lower centerline, third is side.	
IR[10]	[1,99]	5	Unit number for namelist input	I*4
IW[10]	[1,99]	6	Unit number for printed output	I*4
IF[11]	[1,99]	1	Unit number for output of QUICK input data	I*4
	≠IP		Unit IF will be rewound and printed on unit IW	
	=IP		Unit IF will not be rewound or printed. Punch QUICK input data	
IP	[1,99]	7	Unit number for card punch (need not be dummied if not used)	I*4
IPRINT	≠0	0	Cards will be appended to QUICK input (unit IF) to exercise QUICK math model for checking	I*4
	=0		No exercise cards generated	

Table 1. Concluded

Notes:

[1] 'ƀƀƀƀ' = four blank characters.

[2] TH(I) and B(I) constitute segment equation for CURVE(I) ='LINE', i.e., Z(I) = X(I) * TAN(TH(I)) + B(I).

[3] R(I), H(I), and K(I) constitute segment equation for CURVE(I)='ϕGIV', i.e., (X(I)-H(I))**2+(Z(I) -K(I))**2 = R(I)**2.

[4] See Tables 3 and 4 for complete explanation.

[5] Do not input these variables unless rounds, fillets, or slices are present.

[6] XSID(1) and ZSID(1) must both be input as zero. If input at all, at least two values of each must be input.

[7] Variable type: L = logical, I = integer, R = real. Number following * is number of bytes.

[8] When defining a series of segments via X, Z, TH, R, H, K, and B, do not skip an I value when RHϕ(I) is given. RHϕ(I) will fillet/round the intersection of the Ith and I+1st segments. For example, to fillet/round two line segments, input X(1), Z(1), X(2), Z(2), and RHϕ(1). Note further that each fillet/round reduces by one the total number of allowable segments: ten segments with two fillets will overrun QUICK's dimensions, whereas eight segments with two fillets will not. Though fillets/rounds are not treated as segments in the namelist input, they are treated as separate segments in the output after their intersections with adjacent segments have been computed and inserted into the XQ, ZQ, and TQ arrays.

[9] May be negative when TYPE ='ASYM' and if map axis is bent.

[10] If the unit numbers are to be changed, they must also be changed in subroutine INIT because data are read from IR and written on IW before the namelist data are read as namelists.

[11] If IF were set equal to IP(07) in the namelist, the input data would still be read from unit IR(05) copied to unit 01.

Note: Character "ϕ" means capital letter O

"0" means numeric zero 0

[ ] means end points are included; () means end points are excluded.

Table 2. KWIKNOSE Input Options[1] for Three Main Body Lines (ZUCL, ZLCL, YSID)

CURVE(I)	X(I)	Z(I)	H(I)	K(I)	R(I)	SLØPE(I)	TH(I)	B(I)	DIR(I)[3]	May Follow Case 5, 9, 19?	ICASE(I)[2]	EL[2]
'ØGIV'	X	X	—	—	X	—	—	—	X	No	1	582
	X	X	—	—	—	X	—	—	—	No	2[6]	774
	X	—	—	—	X	X	—	—	X	No	3[6]	834
	X	X	—	—	—	—	X	—	—	No	4	526
	—	—	X	X	X	—	—	—	X	Yes	5[4]	624
	X	X	—	—	X	—	X	—	X	Yes	10[5]	590
	X	—	X	X	X	—	—	—	X	Yes	11[5]	626
	—	X	X	X	X	—	—	—	X	Yes	12[5]	628
	—	—	X	X	X	—	X	—	X	Yes	13[5]	632
	X	X	X	X	X	—	X	—	—	Yes	18[5,7]	638
	—	—	—	—	X	X	—	—	X	No	19[6]	832
	X	X	—	—	X	X	—	—	X	Yes	21[5,6]	838
'LINE'	X	X	—	—	—	—	—	—	—	No	6	6
	X	—	—	—	—	—	X	—	—	No	7	10
	—	X	—	—	—	—	X	—	—	No	8	12
	—	—	—	—	—	—	X	X	X	Yes	9[4]	136
	X	X	—	—	—	—	X	—	—	Yes	14[5]	14
	X	—	—	—	—	—	X	X	—	Yes	15[5]	138
	—	X	—	—	—	—	X	X	—	Yes	16[5]	140
	X	X	—	—	—	—	X	X	—	Yes	17[5,7]	142
	X	X	—	—	—	X	—	—	X	No	20[6]	262

L:	512[8]	2[9]	4	16	32	64	256	8	128

[1] One of these options must be chosen for each segment. Insufficient or redundant data will cause execution to be halted.

[2] ICASE(I) and L are not input variables but are used for program control. See subroutines CASE, SEGE, and INTR in Table 8.

[3] DIR(I) is used only when multiple solutions or intersections can occur. When multiple roots are equal, DIR(I) is ignored. See Tables 3 and 4.

[4] These options are not acceptable for last segment of vehicle.

[4,5] Use one of these options if previous segment was given via ICASE=5,9, or 19.

[6] These options are not acceptable for first segment of vehicle.

[7] These options contain redundant data, which if not self consistent to within ERRMAX, will cause execution to halted.

[8] CURVE(I)=ØGIV', L=512; ='LINE; L=0.

[9] If X(I) < 1.E70, L=2; IF X(I) ≥ 1.E70 (default), L=0. Same for remaining variables but with higher powers of 2.

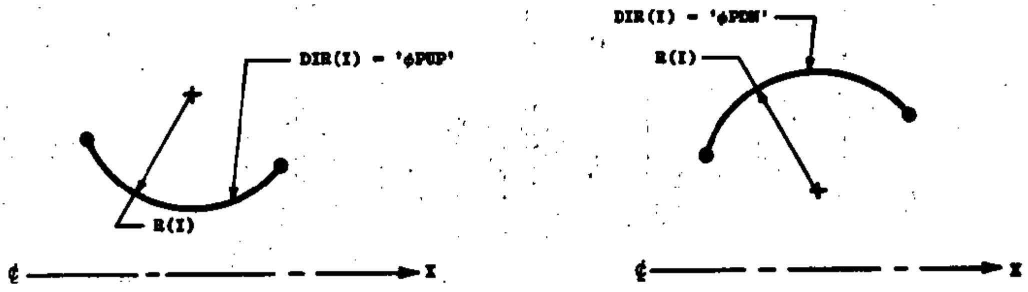
Table 3. Input Description for Variable DIR(I)[1]

ICASE(I)	ICASE(I+1)	ICASE(I-1)	DIR(I)	Meaning
1, 3, 10, 11, 13	—	—	'ΦPUP'[2]	Ogive opens up
			'ΦPDN'[2]	Ogive opens down
5	5	—	'FΦRE'	Use fore solution
			'AFTE'	Use aft solution
			'TΦPS'	Use top solution } Use when X coordinate of fore and aft solutions are equal
			'BΦTS'	Use bottom solution
5	9	—	'FΦRE'	Use fore solution
9	5	—	'AFTE'	Use aft solution
12	—	—	'FΦRE'	Use fore solution
			'AFTE'	Use aft solution
20	—	5	'TΦPS'	Use top solution
			'BΦTS'	Use bottom solution
21	—	5	'ΦPUP'[2]	Ogive opens up
5	21		'ΦPDN'[2]	Ogive opens down
21	—	9	—	DIR(I) not needed
19	—	—	'UPFΦ'	Ogive opens up, use fore solution } For DIR(I) notation, assume 'UP'='ΦPUP', 'DN'='ΦPDN', 'FΦ='FΦRE', 'AF'='AFTE', 'TΦ'='TΦPS', and 'BΦ'='BΦTS' and use Table 4.
			'UPAF'	Ogive opens up, use aft solution
			'UPTΦ'	Ogive opens up, use top solution
			'UPBΦ'	Ogive opens up, use bottom solution
			'DNFΦ'	Ogive opens down, use fore solution
			'DNAF'	Ogive opens down, use aft solution
			'DNTΦ'	Ogive opens down, use top solution
			'DNBΦ'	Ogive opens down, use bottom solution

[1] For the ICASE(I) value, one of the indicated DIR(I) values must be chosen. See Table 4 for clarification.

[2] Except for the noted values, DIR(I) refers to the aft intersection of the Ith segment.

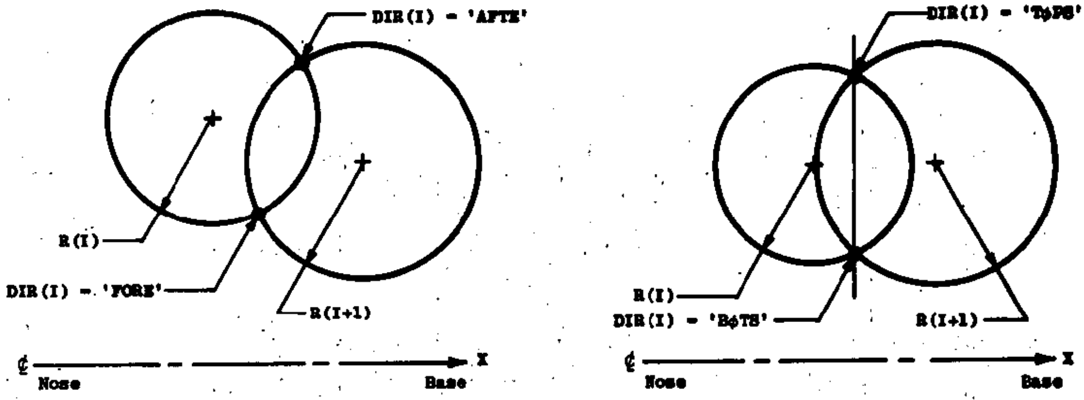
Table 4. Nomenclature for Input Variable DIR(I)



Segment Opens Up

Segment Opens Down

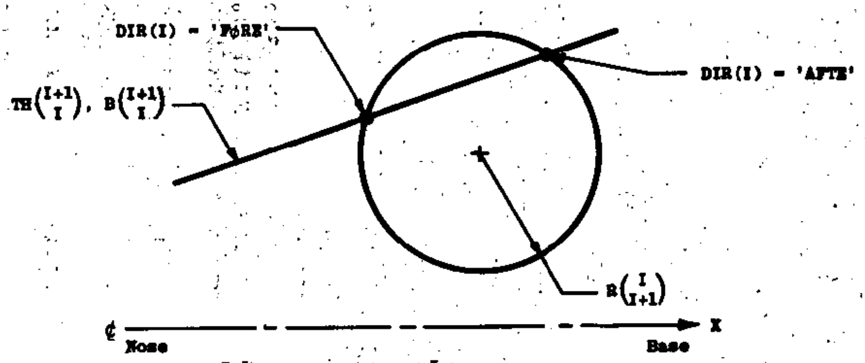
ICASE(I) = 1, 3, 10, 11, 13



Intersections at Different X Stations

Intersections at Same X Station

ICASE(I) = 5 and ICASE(I+1) = 5



$ICASE\binom{I+1}{I} = 5$ and $ICASE\binom{I}{I+1} = 9$; or ICASE(I) = 12

Table 5. Jobstream for Sample Case

```
/*PRIORITY      5
//VKF05333    JOB          (XXX,
//    VRV00146,01,PROJ-NBR),'00000 USER NAME',
//    MSGLEVEL=(2,0),
//    REGION=160K,
//    CLASS=X,
//    TIME=(,15)
// EXEC FORTEPDS,PGMNO=VRV00146
//GO.FT01F001 DD UNIT=WORK,DISP=(NEW,PASS,DELETE),DSN=&&KWIKIN,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),SPACE=(CYL,1)
//GO.FT05F001 DD *
HYPOTHETICAL ASYMMETRICALLY ABLATED, SLICED, BENT CONE
 &INPUT
 TYPE='ASYM',IPRINT=1,
 CURVE=2*'OGIV',2*'LINE',DIR(2)='AFTE',
 H(1)=.5,K(1)=0.,R(1)=.5,TH(1)=45.,H(2)=-.64905,K(2)=1.14905,R(2)=1.125,
 X(3)=4.92,Z(3)=1.84,TH(3)=10.,RHO(2)=.5,X(4)=9.22,Z(4)=2.92218,TH(4)=15.,
 &END
 &INPUT
 CURVE=2*'OGIV',2*'LINE',DIR(2)='AFTE',
 H(1)=.375,K(1)=0.,R(1)=.375,TH(1)=30.,H(2)=-1.8125,K(2)=3.78886,R(2)=4.,
 X(3)=5.08242,Z(3)=1.88,TH(3)=10.,RHO(2)=.5,X(4)=9.22,Z(4)=2.24199,TH(4)=5.,
 &END
 &INPUT
 CURVE=2*'OGIV',3*'LINE',DIR='OPDN','UPAF',
 H(1)=.4375,K(1)=0.,R(1)=.4375,TH(1)=36.325,SLOPE(2)='T',
 R(2)=2.5625,X(3)=4.99909,Z(3)=1.8253,RHO(2)=.5,X(4)=5.00333,Z(4)=1.8253,
 X(5)=9.22,Z(5)=2.57165,,TH(3)=10.,
 XSID=0.,.85535,4.99909,5.00333,9.22,ZSID=0.,.02866,.02866,-.06865,.30026,
 XMAPAX=0.,9.22,ZMAPAX=0.,0.,
 XUSL=6.86,9.65,ZUSL=3.11,1.83,
 XLSL=3.79,7.97,9.94,ZLSL=1.75,1.4,.7,RHOLS(2)=4.,
 XSSL=0.,9.22,YSSL=2.3,2.3,
 &END
// EXEC FORTEPDS,PGMNO=VRV00099,COND=(8,LE)
//GO.FT05F001 DD DSN=&&KWIKIN,DISP=(OLD,DELETE)
//GO.FT07F001 DD UNIT=3330,VOL=SER=666666,DSN=JCAG7115,
//    DCB=(RECFM=U,BLKSIZE=80),DISP=(OLD,KEEP)
/*
```

Table 6. KWIKNOSE Printout for Sample Case

```
INPUT CARDS
-------------------------------------------------------------------------------------
|HYPOTHETICAL ASYMMETRICALLY ABLATED, SLICED, BENT CONE                              |
| &INPUT                                                                            |
| TYPE='ASYM',IPRINT=1,                                                             |
| CURVE=2*'OGIV',2*'LINE',DIR(2)='AFTE',                                            |
| H(1)=.5,K(1)=0.,R(1)=.5,TH(1)=45.,H(2)=-.64905,K(2)=1.14905,R(2)=1.125,           |
| X(3)=4.92,Z(3)=1.84,TH(3)=10.,RHO(2)=.5,X(4)=9.22,Z(4)=2.92218,TH(4)=15.,         |
| &END                                                                              |
| &INPUT                                                                            |
| CURVE=2*'OGIV',2*'LINE',DIR(2)='AFTE',                                            |
| H(1)=.375,K(1)=0.,R(1)=.375,TH(1)=30.,H(2)=-1.8125,K(2)=3.78886,R(2)=4.,          |
| X(3)=5.08242,Z(3)=1.88,TH(3)=10.,RHO(2)=.5,X(4)=9.22,Z(4)=2.24199,TH(4)=5.,       |
| &END                                                                              |
| &INPUT                                                                            |
| CURVE=2*'OGIV',3*'LINE',DIR='OPDN','UPAF',                                        |
| H(1)=.4375,K(1)=0.,R(1)=.4375,TH(1)=36.325,SLOPE(2)='T',                          |
| R(2)=2.5625,X(3)=4.99909,Z(3)=1.8253,RHO(2)=.5,X(4)=5.00333,Z(4)=1.8253,          |
| X(5)=9.22,Z(5)=2.57165,,TH(3)=10.,                                                |
| XSID=0.,.85535,4.99909,5.00333,9.22,ZSID=0.,.02866,.02866,-.06865,.30026,         |
| XMAPAX=0.,9.22,ZMAPAX=0.,0.,                                                      |
| XUSL=6.86,9.65,ZUSL=3.11,1.83,                                                    |
| XLSL=3.79,7.97,9.94,ZLSL=1.75,1.4,.7,RHOLS(2)=4.,                                 |
| XSSL=0.,9.22,YSSL=2.3,2.3,                                                        |
| &END                                                                              |
-------------------------------------------------------------------------------------

.............................................................................................................
INPUT DATA: 1  ZUCL

 I       X             Z            TH            H             K             R             B            RHO                CURVE DIR SLOPE
-------------------------------------------------------------------------------------------------------------------------------------------
 1  9.99999E 69   9.99999E 69   4.50000E 01   5.00000E-01   0.0           5.00000E-01   9.99999E 69   9.99999E 69           OGIV OPDN    F
 2  9.99999E 69   9.99999E 69   9.99999E 69  -6.49050E-01   1.14905E 00   1.12500E 00   9.99999E 69   5.00000E-01           OGIV AFTE    F
 3  4.92000E 00   1.84000E 00   1.00000E 01   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69           LINE OPDN    F
 4  9.22000E 00   2.92218E 00   1.50000E 01   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69           LINE OPDN    F
 5  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN    F
 6  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN    F
 7  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN    F
 8  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN    F
 9  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN    F
10  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN    F

&INPUT1
IR=          1,IW=          6,IF=          1,IP=          7,IPRINT=          1,ERRMAX= 0.49999988E-04
&END
```

Table 6. Continued

```
************************************************************************************************************************

NUMBER OF SEGMENTS NSEG =  4

ICASE( 1) = 13
ICASE( 2) =  5
ICASE( 3) = 14
ICASE( 4) = 14
ICASE( 5) =  0
ICASE( 6) =  0
ICASE( 7) =  0
ICASE( 8) =  0
ICASE( 9) =  0
ICASE(10) =  0

SEGMENT END POINT COORDINATES AND SLOPES:

           X               Z                TA              TF
-------------------------------------------------------------------
 1  0.0             0.0             4.499998E 01    8.999899E 01
 2  1.464467E-01    3.535535E-01    8.524072E 01    4.500000E 01
 3  4.720713E-01    1.055710E 00    9.999997E 00    9.999997E 00
 4  4.919999E 00    1.839999E 00    1.412625E 01    1.412625E 01
 5  9.219999E 00    2.922179E 00    9.999994E 69    9.999994E 69
 6  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69
 7  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69
 8  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69
 9  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69
10  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69
11  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69

SEGMENT EQUATIONS:

           X               H               K               R               M               T               B
--------------------------------------------------------------------------------------------------------------------
 1  1.464467E-01    5.000000E-01    0.0             5.000000E-01    9.999994E 69    4.500000E 01    9.999994E 69
 2  4.720713E-01   -6.490500E-01    1.149050E 00    1.125000E 00    9.999994E 69    9.999994E 69    9.999994E 69
 3  4.919999E 00    9.999994E 69    9.999994E 69    9.999994E 69    1.763268E-01    1.000000E 01    9.724715E-01
 4  9.219999E 00    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    1.500000E 01    4.516907E-01
 5  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69
 6  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69
 7  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69
 8  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69
 9  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69
10  9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69    9.999994E 69

FILLET/ROUND EQUATIONS AND INTERSECTIONS:

          X             KSI            ETA            RHO             X1             Z1             T1             X2             Z2             T2
------------------------------------------------------------------------------------------------------------------------------------------------------------
 1  1.46447E-01    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69
 2  4.72071E-01    8.87869E-01    6.21313E-01    5.00000E-01    4.14969E-01    7.83694E-01    7.10488E 01    8.01045E-01    1.11372E 00    1.00000E 01
 3  4.92000E 00    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69
 4  9.22000E 00    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69
 5  9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69
 6  9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69
 7  9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69
 8  9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69
 9  9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69
10  9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69    9.99999E 69
```

Table 6. Continued

```
FINAL QUICK ARRAY:

          X                Z                TA               TF
------------------------------------------------------------------------
 1  0.0              0.0              4.499998E 01     8.999899E 01
 2  1.464467E-01     3.535535E-01     7.104884E 01     4.500000E 01
 3  4.149692E-01     7.836936E-01     1.000000E 01     7.104884E 01
 4  8.010449E-01     1.113717E 00     9.999997E 00     1.000000E 01
 5  4.919999E 00     1.839999E 00     1.412625E 01     1.412625E 01
 6  9.219999E 00     2.922179E 00     9.999994E 69     9.999994E 69
 7  9.999994E 69     9.999994E 69     9.999994E 69     9.999994E 69
 8  9.999994E 69     9.999994E 69     9.999994E 69     9.999994E 69
 9  9.999994E 69     9.999994E 69     9.999994E 69     9.999994E 69
10  9.999994E 69     9.999994E 69     9.999994E 69     9.999994E 69
.......................................................................................................................
INPUT DATA: 2  ZLCL

 I         X             Z             TH             H              K              A              B             RHO               CURVE DIR SLOPE
-------------------------------------------------------------------------------------------------------------------------------------------------
 1  9.99999E 69   9.99999E 69   3.00000E 01   3.75000E-01   0.0           3.75000F-01   9.99999E 69   9.99999E 69           OGIV OPDN     F
 2  9.99999E 69   9.99999E 69   9.99999E 69  -1.81250E 00   3.78886E 00   4.00000E 00   9.99999E 69   5.00000E-01           OGIV AFTE     F
 3  5.08242E 00   1.88000E 00   1.00000E 01   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69           LINE OPDN     F
 4  9.22000E 00   2.24199E 00   5.00000E 00   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69           LINE OPDN     F
 5  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN     F
 6  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN     F
 7  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN     F
 8  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN     F
 9  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN     F
10  9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69   9.99999E 69                OPDN     F

&INPUT1
IR=            1,IW=           6,IF=            1,IP=           7,IPRINT=           1,ERRMAX= 0.49999988E-04
&END
.......................................................................................................................

NUMBER OF SEGMENTS NSEG =  4

ICASE( 1) = 13
ICASE( 2) =  5
ICASE( 3) = 14
ICASE( 4) = 14
ICASE( 5) =  0
ICASE( 6) =  0
ICASE( 7) =  0
ICASE( 8) =  0
ICASE( 9) =  0
ICASE(10) =  0

SEGMENT END POINT COORDINATES AND SLOPES:

          X                Z                TA               TF
------------------------------------------------------------------------
 1  0.0              0.0              2.999998E 01     8.999899F 01
 2  1.875001E-01     3.247595E-01     4.971005E 01     3.000002E 01
 3  1.238627E 00     1.202235E 00     9.999997E 00     9.999997E 00
 4  5.082419E 00     1.879999E 00     4.999980E 00     4.999980E 00
 5  9.219999E 00     2.241989E 00     9.999994E 69     9.999994E 69
 6  9.999994E 69     9.999994E 69     9.999994E 69     9.999994E 69
 7  9.999994E 69     9.999994E 69     9.999994E 69     9.999994E 69
```

Table 6. Continued

```
 8  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69
 9  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69
10  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69
11  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69

SEGMENT EQUATIONS:

            X             H             K              R             M             T             B
-----------------------------------------------------------------------------------------------------
 1  1.875001E-01  3.750000E-01  0.0           3.750000F-01  9.999994E 69  3.000000E 01  9.999994E 69
 2  1.238627E 00 -1.812500E 00  3.788859E 00  4.000000E 00  9.999994E 69  9.999994E 69  9.999994E 69
 3  5.082419E 00  9.999994E 69  9.999994E 69  9.999994E 69  1.763268E-01  1.000000E 01  9.838324E-01
 4  9.219999E 00  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  5.000000E 00  1.435345E 00
 5  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69
 6  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69
 7  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69
 8  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69
 9  9.999994E 69  9.999994E 69  9.999994E 69  9.999994L 69  9.999994E 69  9.999994E 69  9.999994E 69
10  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69

FILLET/ROUND EQUATIONS AND INTERSECTIONS:

          X            KSI          ETA          RHO           X1           Z1           T1           X2           Z2           T2
---------------------------------------------------------------------------------------------------------------------------------------
 1  1.87500E-01  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 2  1.23863E 00  1.49739E 00  7.40149E-01  5.00000E-01  1.12962E 00  1.07890E 00  4.73521E 01  1.41057E 00  1.23255E 00  1.00000E 01
 3  5.08242E 00  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 4  9.22000E 00  9.9999YE 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 5  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 6  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 7  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 8  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 9  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
10  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69


FINAL QUICK ARRAY:

            X             Z           \ TA            TF
-----------------------------------------------------------
 1  0.0           0.0           2.999998E 01  8.999899E 01
 2  1.875001E-01  3.247595E-01  4.735210E 01  3.000002E 01
 3  1.129624E 00  1.078896E 00  1.000000E 01  4.735210E 01
 4  1.410566E 00  1.232553E 00  9.999997E 00  1.000000E 01
 5  5.082419E 00  1.879999E 00  4.999980E 00  4.999980E 00
 6  9.219999E 00  2.241989E 00  9.999994E 69  9.999994E 69
 7  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69
 8  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69
 9  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69
10  9.999994E 69  9.999994E 69  9.999994E 69  9.999994E 69
.............................................................................................................................
INPUT DATA: 3  YSIO

I        X            Z            TM           H            K            R            B            RHO              CURVE DIR SLOPE
---------------------------------------------------------------------------------------------------------------------------------------
1  9.99999E 69  9.99999E 69  3.63250E 01  4.37500E-01  0.0          4.37500.-01  9.99999E 69  9.99999E 69          OGIV OPDN     F
2  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  2.56250E 00  9.99999E 69  5.00000E-01          OGIV UPAF     T
3  4.99909E 00  1.82530E 00  1.00000E 01  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69          LINE OPDN     F
4  5.00333E 00  1.82530E 00  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69          LINE OPDN     F
5  9.22000E 00  2.57165E 00  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69          LINE OPDN     F
```

Table 6. Continued

```
  6  9.99999E 69  0.0          9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69          OPDN    F
  7  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69          OPDN    F
  8  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69          OPDN    F
  9  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69          OPDN    F
 10  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69          OPDN    F

ZUSL    X-SL          Z-SL          T-SL         RHO-S
-----------------------------------------------------------
  1  6.86000E 00  3.11000E 00  9.99999E 69  9.99999E 69
  2  9.65000E 00  1.83000E 00  9.99999E 69  9.99999E 69
  3  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  4  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  5  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  6  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  7  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  8  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  9  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 10  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69

ZLSL    X-SL          Z-SL          T-SL         RHO-S
-----------------------------------------------------------
  1  3.79000E 00  1.75000E 00  9.99999E 69  9.99999E 69
  2  7.97000E 00  1.40000E 00  9.99999E 69  4.00000E 00
  3  9.94000E 00  7.00000E-01  9.99999E 69  9.99999E 69
  4  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  5  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  6  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  7  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  8  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  9  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 10  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69

YSSL    X-SL          Z-SL          T-SL         RHO-S
-----------------------------------------------------------
  1  0.0          2.30000E 00  9.99999E 69  9.99999E 69
  2  9.22000E 00  2.30000E 00  9.99999E 69  9.99999E 69
  3  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  4  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  5  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  6  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  7  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  8  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  9  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 10  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69

       XMAPAX        ZMAPAX         XSID          ZSID
-----------------------------------------------------------
  1  0.0          0.0          0.0          0.0
  2  9.22000E 00  0.0          8.55350E-01  2.86600E-02
  3  9.99999E 69  9.99999E 69  4.99909E 00  2.86600E-02
  4  9.99999E 69  9.99999E 69  5.00333E 00 -6.86499E-02
  5  9.99999E 69  9.99999E 69  9.22000E 00  3.00260E-01
  6  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  7  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  8  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
  9  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 10  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
```

Table 6. Continued

```
&INPUT1
IR=         1,IW=         6,IF=         1,IP=         7,IPRINT=         1,ERRMAX= 0.49999988E-04
&END
***************************************************************************************************************

NUMBER OF SEGMENTS NSEG =  5

ICASE( 1) = 13
ICASE( 2) = 19
ICASE( 3) = 14
ICASE( 4) =  6
ICASE( 5) =  6
ICASE( 6) =  0
ICASE( 7) =  0
ICASE( 8) =  0
ICASE( 9) =  0
ICASE(10) =  0

SEGMENT END POINT COORDINATES AND SLOPES:

         X               Z              TA              TF
------------------------------------------------------------------
 1  0.0            0.0            3.632498E 01   8.999899: 01
 2  1.783406E-01   3.524806E-01   5.893269E 01   3.632498E 01
 3  8.553475E-01   1.094646E 00   9.999994E 00   9.999994E 00
 4  4.999089E 00   1.825299E 00   0.0            0.0
 5  5.003329E 00   1.825299E 00   1.003739E 01   1.003739E 01
 6  9.219999E 00   2.571650E 00   9.999994E 69   9.999994E 69
 7  9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69
 8  9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69
 9  9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69
10  9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69
11  9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69

SEGMENT EQUATIONS:

         X               M               K               R               H               T               B
----------------------------------------------------------------------------------------------------------------
 1  1.783406E-01   4.375000E-01   0.0            4.375000:-01   9.999994E 69   3.632500E 01   9.999994E 69
 2  8.553475E-01  -1.339593E 00   2.417009E 00   2.562500E 00   9.999994E 69   9.999994E 69   9.999994E 69
 3  4.999089E 00   9.999994E 69   9.999994E 69   9.999994E 69   1.763268E-01   1.000000E 01   9.438258E-01
 4  5.003329E 00   9.999994E 69   9.999994E 69   9.999994E 69   0.0            0.0            1.825299E 00
 5  9.219999E 00   9.999994E 69   9.999994E 69   9.999994E 69   1.769999E-01   1.003739E 01   9.397106E-01
 6  9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69
 7  9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69
 8  9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69
 9  9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69
10  9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69   9.999994E 69

FILLET/ROUND EQUATIONS AND INTERSECTIONS:

        X            KSI          ETA          RHO           X1           Z1           T1           X2           Z2           T2
---------------------------------------------------------------------------------------------------------------------------------------
 1  1.78341E-01  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 2  8.55347E-01  1.15439E 00  6.39662E-01  5.00000E-01  7.47208E-01  9.29842E-01  5.45243E 01  1.06756E 00  1.13207E 00  1.00000E 01
 3  4.99909E 00  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 4  5.00333E 00  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 5  9.22000E 00  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 6  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 7  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
```

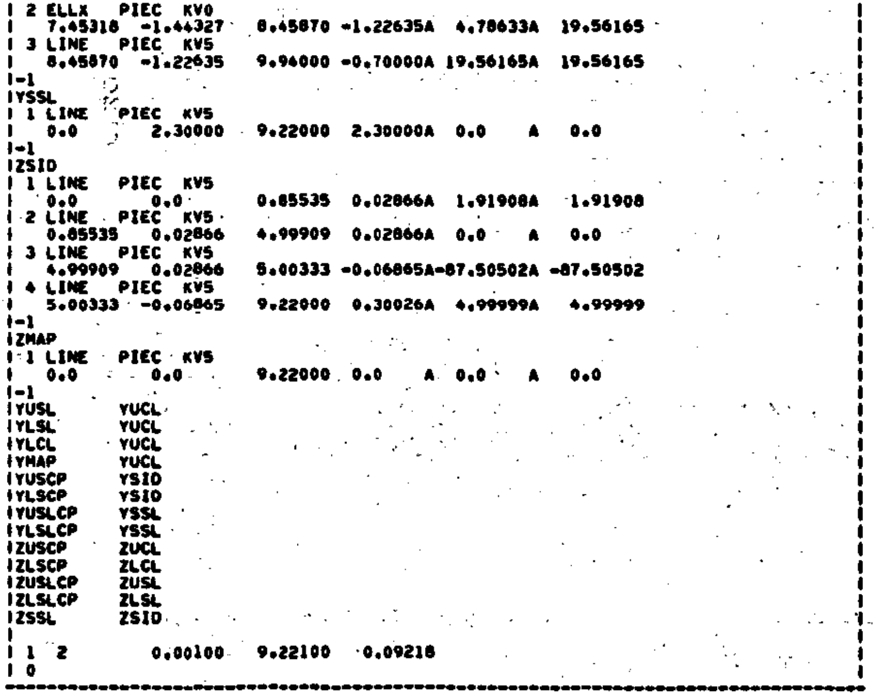
Table 6. Continued

```
 8  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
 9  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69
10  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69  9.99999E 69

FINAL QUICK ARRAY:

          X                Z                TA               TF
-------------------------------------------------------------------
 1  0.0             0.0             3.632498E 01     8.999899E 01
 2  1.783406E-01    3.524806E-01    5.452428E 01     3.632498E 01
 3  7.472076E-01    9.298420E-01    1.000000E 01     5.452428E 01
 4  1.067565E 00    1.132066E 00    9.999994E 00     1.000000E 01
 5  4.999089E 00    1.825299E 00    0.0              0.0
 6  5.003329E 00    1.825299E 00    1.003739E 01     1.003739E 01
 7  9.219999E 00    2.571650E 00    9.999994E 69     9.999994E 69
 8  9.999994E 69    9.999994E 69    9.999994E 69     9.999994E 69
 9  9.999994E 69    9.999994E 69    9.999994E 69     9.999994E 69
10  9.999994E 69    9.999994E 69    9.999994E 69     9.999994E 69

SLICE GEOMETRY:

                        UPPER                                          LOWER                                          SIDE
          X               Z                T                X            Z               T                X            Y               T
----------------------------------------------------------------------------------------------------------------------------------------------
 1  0.0             6.257238E 00 -2.464478E 01    0.0             2.067344E 00 -4.786331E 00    0.0             2.299999E 00  0.0
 2  9.650000E 00    1.830000E 00 -2.464478E 01    7.969999E 00    1.400000E 00 -4.786331E 00    9.219999E 00    2.299999E 00  0.0
 3  9.999994E 69    9.999994E 69  9.999994E 69    9.940000E 00    7.000000E-01 -1.956165E 01    9.999994E 69    9.999994E 69  9.999994E 69
 4  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69
 5  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69
 6  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69
 7  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69
 8  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69
 9  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69
10  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69

SLICE ROUND/FILLET CENTERS AND INTERSECTIONS:

         BI            BJ            A3            A4            X1            Z1            T1            X2            Z2            T2
----------------------------------------------------------------------------------------------------------------------------------------------

22 -1.94665E 00 -1.30367E-02 -2.54278E 00  7.11942E 00  7.45318E 00  1.44327E 00 -4.78633E 00  8.45870E 00  1.22635E 00 -1.95616E 01


FINAL QUICK ARRAYS FOR ROUNDED/FILLETTED SLICES:

                        UPPER                                          LOWER                                          SIDE
          X               Z                T                X            Z               T                X            Y               T
----------------------------------------------------------------------------------------------------------------------------------------------
 1  0.0             6.257238E 00 -2.464478E 01    0.0             2.067344E 00 -4.786331E 00    0.0             2.299999E 00  0.0
 2  9.650000E 00    1.830000E 00 -2.464478E 01    7.453176E 00    1.443274E 00 -4.786331E 00    9.219999E 00    2.299999E 00  0.0
 3  9.999994E 69    9.999994E 69  9.999994E 69    8.458698E 00    1.226351E 00 -1.956165E 01    9.999994E 69    9.999994E 69  9.999994E 69
 4  9.999994E 69    9.999994E 69  9.999994E 69    9.940000E 00    7.000000E-01 -1.956165E 01    9.999994E 69    9.999994E 69  9.999994E 69
 5  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69
 6  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69
 7  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69
 8  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69
 9  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69
10  9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69    9.999994E 69    9.999994E 69  9.999994E 69

LISTING OF QUICK INPUT DECK ON UNIT  1:
```

Table 6. Continued

```
--------------------------------------------------------------------------------------------
|HYPOTHETICAL ASYMMETRICALLY ABLATED, SLICED, BENT CONE
| 1
| 1 6       NONCIRCULAR CROSS SECTION
|LELL      1ELLI  PIEC     BLCL        SID        LSCP
|UELL      2ELLI  PIEC      SID       TUCL        USCP
|LBSL      3LINE  PIEC     BLSL        LSLCP                                 LELL
|LSSL      4LINE  PIEC      LSLCP      SSL                        LELL
|USSL      5LINE  PIEC      SSL        USLCP                                 UELL
|UTSL      6LINE  PIEC      USLCP     TUSL                        UELL
| 1         MAP
| 1 1          0.0        9.22000
|YUCL
| 1 LINE    PIEC  KV5
|   0.0        0.0        9.22000  0.0     A  0.0     A   0.0
|-1
|ZUCL
| 1 ELLX    PIEC  KV0
|   0.0        0.0        0.14645  0.35355A 89.99899A  44.99998
| 2 ELLX    PIEC  KV0
|   0.14645   0.35355     0.41497  0.78369A 45.00000A  71.04884
| 3 ELLX    PIEC  KV0
|   0.41497   0.78369     0.80104  1.11372A 71.04884A  10.00000
| 4 LINE    PIEC  KV5
|   0.80104   1.11372     4.92000  1.84000A 10.00000A  10.00000
| 5 LINE    PIEC  KV5
|   4.92000   1.84000     9.22000  2.92218A 14.12625A  14.12625
|-1
|ZLCL
| 1 ELLX    PIEC  KV0
|   0.0        0.0        0.18750 -0.32476A-89.99899A -29.99998
| 2 ELLX    PIEC  KV0
|   0.18750  -0.32476     1.12962 -1.07890A-30.00002A -47.35210
| 3 ELLX    PIEC  KV0
|   1.12962  -1.07890     1.41057 -1.23255A-47.35210A -10.00000
| 4 LINE    PIEC  KV5
|   1.41057  -1.23255     5.08242 -1.88000A-10.00000A -10.00000
| 5 LINE    PIEC  KV5
|   5.08242  -1.88000     9.22000 -2.24199A -4.99998A  -4.99998
|-1
|YSID
| 1 ELLX    PIEC  KV0
|   0.0        0.0        0.17834  0.35248A 89.99899A  36.32498
| 2 ELLX    PIEC  KV0
|   0.17834   0.35248     0.74721  0.92984A 36.32498A  54.52428
| 3 ELLX    PIEC  KV0
|   0.74721   0.92984     1.06756  1.13207A 54.52428A  10.00000
| 4 LINE    PIEC  KV5
|   1.06756   1.13207     4.99909  1.82530A 10.00000A   9.99999
| 5 LINE    PIEC  KV5
|   4.99909   1.82530     5.00333  1.82530A  0.0     A   0.0
| 6 LINE    PIEC  KV5
|   5.00333   1.82530     9.22000  2.57165A 10.03739A  10.03739
|-1
|ZUSL
| 1 LINE    PIEC  KV5
|   0.0        6.25724    9.65000  1.83000A-24.64478A -24.64478
|-1
|ZLSL
| 1 LINE    PIEC  KV5
|   0.0       -2.06734    7.45318 -1.44327A  4.78633A   4.78633
```

Table 6. Concluded



```
| 2 ELLX   PIEC  KV0
|   7.45318  -1.44327   8.45870 -1.22635A  4.78633A  19.56165
| 3 LINE   PIEC  KV5
|   8.45870  -1.22635   9.94000 -0.70000A 19.56165A  19.56165
|-1
|YSSL
| 1 LINE   PIEC  KV5
|   0.0       2.30000   9.22000  2.30000A  0.0    A   0.0
|-1
|ZSID
| 1 LINE   PIEC  KV5
|   0.0       0.0       0.85535  0.02866A  1.91908A   1.91908
| 2 LINE   PIEC  KV5
|   0.85535   0.02866   4.99909  0.02866A  0.0    A   0.0
| 3 LINE   PIEC  KV5
|   4.99909   0.02866   5.00333 -0.06865A-87.50502A -87.50502
| 4 LINE   PIEC  KV5
|   5.00333  -0.06865   9.22000  0.30026A  4.99999A   4.99999
|-1
|ZMAP
| 1 LINE   PIEC  KV5
|   0.0       0.0       9.22000  0.0    A  0.0    A   0.0
|-1
|YUSL      YUCL
|YLSL      YUCL
|YLCL      YUCL
|YMAP      YUCL
|YUSCP     YSID
|YLSCP     YSID
|YUSLCP    YSSL
|YLSLCP    YSSL
|ZUSCP     ZUCL
|ZLSCP     ZLCL
|ZUSLCP    ZUSL
|ZLSLCP    ZLSL
|ZSSL      ZSID
|
| 1  2        0.00100   9.22100   0.09218
| 0
-----------------------------------------------------------------------------------
```

Table 7. Definitions of Logical Names for QUICK Cross-Section and Body Line Models

Cross-Section Models

Logical Name	Definition
LELL	Lower ellipse
UELL	Upper ellipse
UTSL	Upper top slicing plane
USSL	Upper side slicing plane
LSSL	Lower side slicing plane
LBSL	Lower bottom slicing plane

Body Line Models

Logical Name	Definition
LCL	Lower centerline
LSCP	Lower slope control point
MAPAX	Map axis
SID	Side
UCL	Upper centerline
USCP	Upper slope control point
LSLCP	Control point for lower horizontal and vertical slicing planes
LSL	Centerline on lower slice
SSL	Side on side slice
USL	Centerline on upper slice
USLCP	Control point for upper horizontal and vertical slicing planes

Note: See Fig. 8 also.

Table 8. KWIKNOSE Source Listing

```
C KWIKNOSE - PROGRAM TO SET UP QUICK INPUT FOR AN AXISMMETRIC STACK OF
C CONES AND OGIVES. NOT NECESSARILY TANGENT
C ORIGINAL PROGRAM VERSION 2/18/77
C---------------------------------------------------------------------
C INPUT INSTRUCTIONS (SEE AEDC-TR-77-89 FOR DETAILS)
C FIRST INPUT CARD IS ID DATA (80A1,QUICK READS ONLY 60A1)
C REMAINING INPUT IS VIA NAMELIST /INPUT/
C SET IPRINT = 1 TO MAKE QUICK EXERCISE MATH MODEL
C PROVIDE DD CARD FOR UNIT IF (DEFAULT = 1) WITH DCB=(RECFM=FB,
C LRECL=80), 1 CYL LONG SEQUENTIAL FILE
C IF IP=IF QUICK INPUT DECK WILL BE WRITTEN ON IP WITHOUT REWINDING. USE
C TO PUNCH.
C---------------------------------------------------------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
      LB=1
      CALL INIT
      CALL INPT
      IF(TYPE.EQ. ASYM)GO TO 100
      CALL CASE
      CALL SEGE
      CALL INTR
      CALL QARY
      CALL PRNT
      CALL ROFL
      CALL SLCE
      CALL RFSL
      CALL QUCK
      CALL LIST
      STOP
  100 DO 120 LB=1,3
      IF(LB.EQ.1)GO TO 110
      CALL INIT
      CALL INPT1
  110 CALL CASE
      CALL SEGE
      CALL INTR
      CALL QARY
      CALL PRNT
      CALL ROFL
      CALL COPY
```

Table 8. Continued

```
  120 CONTINUE
      CALL SLCE
      CALL RFSL
      DO 130 LB=4,8
      CALL COPY
  130 CONTINUE
      CALL BLCK
      CALL OK3D
      CALL LIST
      STOP
      END
C***********************************************************************
      SUBROUTINE INIT
C-------------------
C INITIALIZE VARIABLES
C-------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSIO(10),ZSIO(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,A1,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
 1000 CONTINUE
      IF(LB.GT.1)GO TO 1100
      IPRINT=0
      IR=5
      IW=6
      IF=1
      IP=7
      ERRMAX=5.E-5
 1100 NSEG=10
      PIO180=3.1415926/180.
      ISTOP=0
      JSTOP=0
      KSTOP=0
      RETURN
      END
C***********************************************************************
      SUBROUTINE INPT
C-----------------
C OBTAIN INPUT DATA
C-----------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
```

Table 8. Continued

```
    3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
    4,VB(10,3,2,8),CRVB  (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
    5 BLB(8)
     COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
     COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
    1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
    2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
    3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
     COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
    1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
     COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
     COMMON /LS/TI,T
     REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
     LOGICAL*1 SLOPE,IDEN,CARD,TI,T
     DIMENSION  XUSL(10),ZUSL(10),TUSL(10),
    1           XLSL(10),ZLSL(10),TLSL(10),
    2           XSSL(10),YSSL(10),TSSL(10)
     EQUIVALENCE
    1 (V(1,1,1),XUSL(1)),(V(1,2,1),ZUSL(1)),(V(1,3,1),TUSL(1))
    2,(V(1,1,2),XLSL(1)),(V(1,2,2),ZLSL(1)),(V(1,3,2),TLSL(1))
    3,(V(1,1,3),XSSL(1)),(V(1,2,3),YSSL(1)),(V(1,3,3),TSSL(1))
     DIMENSION RHOUS(10),RHOLS(10),RHOSS(10)
     EQUIVALENCE (RHOSL(1,1),RHOUS(1))
    1           ,(RHOSL(1,2),RHOLS(1))
    2           ,(RHOSL(1,3),RHOSS(1))
     NAMELIST/INPUT/X,Z,TH,CURVE,R,H,K,  B,IR,IW,IF,ERRMAX,DIR,IPRINT,
    1 IP,RHO,SLOPE,XUSL,ZUSL,TUSL,XLSL,ZLSL,TLSL,XSSL,YSSL,TSSL
    2,RHOUS,RHOLS,RHOSS,TYPE,XMAPAX,ZMAPAX,XSID,ZSID
     NAMELIST/INPUT1/IR,IW,IF,IP,IPRINT,ERRMAX
2000 CONTINUE
     IF(IF.EQ.IP)GO TO 2005
     REWIND IF
2002 READ(IR,90001,END=2004)CARD
     WRITE(IF,90001,END=2003)CARD
     GO TO 2002
2003 STOP 2003
2004 IR=IF
     REWIND IF
2005 CONTINUE
2010 FORMAT(80A1)
     WRITE(IW,2001)
2001 FORMAT('1',30X,23('<*>')/31X,23('<*>')/
     - 31X,'<*>',63X,'<*>'/
     -                                    31X,'<*>  PROGRAM KWIKNOSE
     1 TO SET UP QUICK INPUT FOR NOSE GEOMETRIES  <*>'/
     2 31X,'<*>',63X,'<*>'/
     3 31X,'<*>  VON KARMAN GAS DYNAMICS FACILITY (VKF/ADP)      ',13X,
     4 '<*>'/
     5 31X,'<*>',63X,'<*>'/
     6 31X,'<*>  ARNOLD ENGINEERING DEVELOPMENT CENTER, TENNESSEE 37389'
     7, 7X,'<*>'/
     8 31X,'<*>',63X,'<*>'/
     9 31X,23('<*>')/31X,23('<*>')////)
      IF(IF.EQ.IP)GO TO 2013
      REWIND IF
      WRITE(IW,2006)
```

Table 8. Continued

```
2006 FORMAT(' INPUT CARDS')
     WRITE(IW,2007)
2007 FORMAT(' ',82('='))
2008 READ(IF,90001,END=2011)CARD
     WRITE(IW,2009)CARD
2009 FORMAT(' |',80A1,'|')
     GO TO 2008
2011 WRITE(IW,2007)
2013 CONTINUE
     IF(IF.NE.IP)REWIND IF
     READ(IR,2010,END=2020)IDEN
     READ(IR,INPUT,END=2040)
2019 WRITE(IW,9050)
     WRITE(IW,2111)LB,BLB(LB) .
2111 FORMAT(' INPUT DATA:',I2,2X,A4)
     WRITE(IW,2112)
2112 FORMAT('0 I', 6X,' X',11X,' Z',11X,'TH',11X,' H',11X,' K',11X,
    1 ' R',11X,' B',11X,'RHO')
     WRITE(IW,2113)
2113 FORMAT('+',117X,'CURVE DIR SLOPE')
     WRITE(IW,2114)
2114 FORMAT(' ',132('-'))
     WRITE(IW,2115)(I,X(I),Z(I),TH(I),H(I),K(I),R(I),B(I),RHO(I),
    1CURVE(I),DIR(I),SLOPE(I),
    2 I=1,10)
2115 FORMAT(10(' ',I2,1P8E13.5,11X,2(1X,A4),4X,A1/))
     IF(TYPE.EQ.ASYM.AND.LB.LT.3)GO TO 2023
     DO 2118 L=1,3
     WRITE(IW,2116)SL(L)
2116 FORMAT('0',A4, 3X,'X-SL', 9X,'Z-SL', 9X,'T-SL', 8X,'RHO-S'/
    1 ' ',54('-'))
     WRITE(IW,2117)(I,(V(I,J,L),J=1,3),RHOSL(I,L),I=1,10)
2117 FORMAT(10(' ',I2,1P4E13.5/))
2118 CONTINUE
     WRITE(IW,2021)
2021 FORMAT('0',6X,'XMAPAX',7X,'ZMAPAX',8X,'XSID',9X,'ZSID'
    1/' ',54('-'))
     WRITE(IW,2022)(I,XMAPAX(I),ZMAPAX(I),XSID(I),ZSID(I),I=1,10)
2022 FORMAT(10(' ',I2,1P4E13.5/))
2023 WRITE(IW,INPUT1)
     WRITE(IW,9050)
     GO TO 3000
2020 WRITE(IW,2030)
2030 FORMAT(' COND CODE 2020: MISSING ID CARD')
     WRITE(IW,2035)
2035 FORMAT('0EXECUTION HALTED')
     STOP 2020
2040 WRITE(IW,2050)IR
2050 FORMAT(' COND CODE 2040: EOF ON UNIT IR=',I2)
     WRITE(IW,2035)
     STOP 2040
3000 RETURN
     ENTRY INPT1
     IF(IF.NE.IP)IR=IF
     READ(IR,INPUT,END=2040)
     GO TO 2019
```

Table 8. Continued

```
 9050 FORMAT(' ',132('.'))
90001 FORMAT(80A1)
90002 FORMAT('1')
      END
C***********************************************************************
      SUBROUTINE CASE
C---------------
C DETERMINE ICASE
C---------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
      DIMENSION E(10,7)
      EQUIVALENCE (E(1,1),X (1))
     1            ,(E(1,2),Z (1))
     2            ,(E(1,3),TH(1))
     3            ,(E(1,4),H (1))
     4            ,(E(1,5),K (1))
     5            ,(E(1,6),R (1))
     6            ,(E(1,7),B (1))
 3000 CONTINUE
C DETERMINE NO. SEGMENTS NSEG
      DO 3010 J=1,10
      I=10-J+1
      ICASE(I)=0
      IF(CURVE(I).NE.BLNK)GO TO 3010
      NSEG=I-1
 3010 CONTINUE
      WRITE(IW,3015)NSEG
 3015 FORMAT('0NUMBER OF SEGMENTS NSEG =',I3)
C MAKE SURE CURVE='LINE' OR 'OGIV' SPECIFIED FOR EACH SEG
      DO 3040 I=1,NSEG
      DO 3020 J=1,2
      IF(CURVE(I).EQ.KRV(J))GO TO 3040
 3020 CONTINUE
      WRITE(IW,3030)I,CURVE(I),KRV(1),KRV(2)
 3030 FORMAT(' COND CODE 3030: CURVE(',I2,')=',A4,' NOT = TO ',A4,' OR '
     1,A4)
      ISTOP=1
 3040 CONTINUE
      IF(ISTOP.EQ.1)STOP 3030
      DO 3080 I=1,NSEG
```

Table 8. Continued

```
      L=0
      DO 3050 J=1,7
      IF(E(I,J).LT.RINF)L=L+2**J
 3050 CONTINUE
      IF(SLOPE(I).EQ.T)L=L+256
      IF(CURVE(I).EQ.KRV(2))L=L+512
      DO 3060 J=1,21
      IF(L.NE.JCASE(J))GO TO 3060
      ICASE(I)=J
      GO TO 3080
 3060 CONTINUE
      ISTOP=1
      WRITE(IW,3070)I
 3070 FORMAT(' COND CODE 3070: INPUT DATA FOR SEGMENT ',I2,' DOES NOT MA
     1TCH ANY OF PROGRAMMED OPTIONS')
 3080 CONTINUE
      WRITE(IW,3090)(I,ICASE(I),I=1,10)
 3090 FORMAT(10(/' ICASE(',I2,') =',I3))
      IF(ISTOP.EQ.0)GO TO 4000
      WRITE(IW,2035)
      STOP 3070
 4000 RETURN
 2035 FORMAT('0EXECUTION HALTED')
      END
C*********************************************************************
      SUBROUTINE SEGE
C--------------------------
C DETERIMINE SEGMENT EQUATION
C--------------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BL8(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,L8,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CAHD,TI,T
 4000 CONTINUE
      XJ=0.
      ZJ=0.
      DO 4990 I=1,NSEG
      I1=ICASE(I)
      IF(I.EQ.1)GO TO 4007
      IF(ICASE(I-1).NE.5.AND.ICASE(I-1).NE.9)GO TO 4007
      DO 4006 J=1,7
      IF(I1.NE.NCASE(J))GO TO 4006
      WRITE(IW,4005)ICASE(I),ICASE(I-1)
```

Table 8. Continued

```
 4005 FORMAT(' COND CODE 4005: CASE ',I2,' MAY NOT BE USED AFTER CASE ',
     1I2)
      ISTOP=1
      GO TO 4007
 4006 CONTINUE
 4007 CONTINUE
      XI=X(I)
      ZI=Z(I)
      MI=M(I)
      KI=K(I)
      RI=R(I)
      BI=B(I)
      TI=SLOPE(I)
      THI=TH(I)
C              1    2    3    4    5    6    7    8    9   10
C             11   12   13   14   15   16   17   18   19   20   21
      GO TO(4070,4090,4115,4120,4150,4010,4020,4030,4150,4160,
     1      4170,4180,4190,4200,4300,4400,4410,4420,4430,4460,4490
     2),II
C ICASE=6
 4010 M(I)=(ZI-ZJ)/(XI-XJ)
      TH(I)=ATAN(M(I))/PIO180
      B(I)=ZI-M(I)*XI
      GO TO 4040
C ICASE=7
 4020 M(I)=TAN(TH(I)*PIO180)
      Z(I)=M(I)*(X(I)-XJ)+ZJ
      B(I)=Z(I)-M(I)*XI
      GO TO 4040
C ICASE=8
 4030 M(I)=TAN(TH(I)*PIO180)
      X(I)=XJ+(ZI-ZJ)/M(I)
      B(I)=ZI-M(I)*X(I)
      GO TO 4040
C CHECK REDUNDANT INPUT DATA
 4040 JSTOP=0
      IF(XI.LT.RINF.AND.ABS(XI-X(I)).GT.ERRMAX)JSTOP=1
      IF(JSTOP.EQ.1)WRITE(IW,4050)JSTOP,I,ERRMAX
      IF(ZI.LT.RINF.AND.ABS(ZI-Z(I)).GT.ERRMAX)JSTOP=2
      IF(JSTOP.EQ.2)WRITE(IW,4050)JSTOP,I,ERRMAX
      IF(BI.LT.RINF.AND.ABS(BI-B(I)).GT.ERRMAX)JSTOP=3
      IF(JSTOP.EQ.3)WRITE(IW,4050)JSTOP,I,ERRMAX
      IF(THI.LT.RINF.AND.ABS(THI-TH(I)).GT.ERRMAX)JSTOP=4
      IF(JSTOP.EQ.4)WRITE(IW,4050)JSTOP,I,ERRMAX
 4050 FORMAT(' COND CODE 4050-',I1,': REDUNDANT INPUT DATA FOR SEGMENT '
     1,I2,' NOT SELF CONSISTENT TO WITHIN ',1PE15.7)
      IF(JSTOP.NE.0)WRITE(IW,4055)
 4055 FORMAT(' SUGGEST USER CHECK IS INPUT DATA')
      GO TO 4150
C ICASE=1
 4070 MJ=-(XI-XJ)/(ZI-ZJ)
      BJ=ZI+ZJ-MJ*(XI+XJ)
      BJ=0.5*BJ
      A1=(XI+MJ*(ZI-BJ))/(1.+MJ**2)
      A2= (XI+MJ*(ZI-BJ))**2-(1.+MJ**2)*(XI**2+(ZI-BJ)**2-RI**2)
      A2=SQRT(A2)/(1.+MJ**2)
```

Table 8. Continued

```
      H(I)=A1+A2
      K(I)=MJ*H(I)+BJ
      IF(DIR(I).EQ.OPDN.AND.K(I).LT.ZI)GO TO 4080
      H(I)=A1-A2
      K(I)=MJ*H(I)+BJ
 4080 M(I)=-(X(I)-H(I))/(Z(I)-K(I))
      TH(I)=ATAN(M(I))/PIO180
      GO TO 4130
C ICASE=2
 4090 IF(I.GT.1)GO TO 4110
      WRITE(IW,4100)
 4100 FORMAT(' COND CODE 4100: FIRST SEGMENT MAY NOT BE SPECIFIED TANGEN
     1T TO PREVIOUS SEGMENT')
      WRITE(IW,2035)
      STOP 4100
 4110 MJ=-(XI-XJ)/(ZI-ZJ)
      BJ=ZI+ZJ-MJ*(XI+XJ)
      BJ=0.5*BJ
      MI=-1./TAN(PIO180*TH(I-1))
      BI=ZJ-MI*XJ
      H(I)=-(BJ-BI)/(MJ-MI)
      K(I)=MJ*H(I)+BJ
      R(I)=SQRT((XI-H(I))**2+(ZI-K(I))**2)
      TH(I)=-ATAN((XI-H(I))/(ZI-K(I)))/PIO180
      GO TO 4130
C ICASE=3
 4115 S=1.
      IF(DIR(I).NE.OPDN)S=-1.
      H(I)=XJ+RI*SIN(PIO180*THJ)*S
      K(I)=ZJ-RI*COS(PIO180*THJ)*S
      Z(I)=K(I)+S*SQRT(RI**2-(XI-H(I))**2)
      TH(I)=-ATAN((XI-H(I))/(Z(I)-K(I)))/PIO180
      GO TO 4130
C ICASE=4
 4120 MJ=-1./TAN(PIO180*TH(I))
      BJ=ZI-MJ*XI
      MIJ=-(XI-XJ)/(ZI-ZJ)
      BIJ=ZI+ZJ-MIJ*(XI+XJ)
      BIJ=0.5*BIJ
      H(I)=-(BJ-BIJ)/(MJ-MIJ)
      K(I)=MJ*H(I)+BJ
      R(I)=SQRT((XI-H(I))**2+(ZI-K(I))**2)
      GO TO 4130
C CHECK REDUNDANT DATA
 4130 KSTOP=0
      IF(XI .LT.RINF.AND.ABS(XI -X (I)).GT.ERRMAX)KSTOP=1
      IF(KSTOP.EQ.1)WRITE(IW,4140)KSTOP,I,ERRMAX
      IF(ZI .LT.RINF.AND.ABS(ZI -Z (I)).GT.ERRMAX)KSTOP=2
      IF(KSTOP.EQ.2)WRITE(IW,4140)KSTOP,I,ERRMAX
      IF(THI.LT.RINF.AND.ABS(THI-TH(I)).GT.ERRMAX)KSTOP=3
      IF(KSTOP.EQ.3)WRITE(IW,4140)KSTOP,I,ERRMAX
      IF(HI .LT.RINF.AND.ABS(HI -H (I)).GT.ERRMAX)KSTOP=4
      IF(KSTOP.EQ.4)WRITE(IW,4140)KSTOP,I,ERRMAX
      IF(KI .LT.RINF.AND.ABS(KI -K (I)).GT.ERRMAX)KSTOP=5
      IF(KSTOP.EQ.5)WRITE(IW,4140)KSTOP,I,ERRMAX
      IF(RI .LT.RINF.AND.ABS(RI -R (I)).GT.ERRMAX)KSTOP=6
```

Table 8. Continued

```
      IF(KSTOP.EQ.6)WRITE(IW,4140)KSTOP,I,ERRMAX
 4140 FORMAT(' COND CODE 4140-',I1,': REDUNDANT INPUT DATA FOR SEGMENT '
     1,I2,' NOT SELF CONSISTENT TO WITHIN ',1PE15.7)
      IF(KSTOP.NE.0)WRITE(IW,4055)
      GO TO 4150
 4150 CONTINUE
      XJ=X(I)
      ZJ=Z(I)
      THJ=TH(I)
      GO TO 4990
C ICASE=10
 4160 S=1.
      IF(DIR(I).EQ.OPUP)S=-1.
      H(I)=XI+S*RI*SIN(PIO180*THI)
      K(I)=ZI-S*RI*COS(PIO180*THI)
      GO TO 4130
C ICASE=11
 4170 S=1.
      IF(DIR(I).EQ.OPUP)S=-1.
      Z(I)=KI+S*SQRT(RI**2-(XI-HI)**2)
      TH(I)=-ATAN((XI-HI)/(Z(I)-KI))/PIO180
      GO TO 4130
C ICASE=12
 4180 S=1.
      IF(DIR(I).EQ.FORE.OR.DIR(I).EQ.AFTE)GO TO 4186
      WRITE(IW,4185)I
 4185 FORMAT(' COND CODE 4185: DIR MUST BE EITHER ''FORE'' OR ''AFTE'' F
     1ORE SEGMENT',I3)
      WRITE(IW,2035)
      STOP 4185
 4186 CONTINUE
      IF(DIR(I).EQ.FORE)S=-1.
      X(I)=HI+S*SQRT(RI**2-(ZI-KI)**2)
      TH(I)=-ATAN((X(I)-HI)/(ZI-KI))/PIO180
      GO TO 4130
C ICASE=13
 4190 S=1.
      IF(DIR(I).EQ.OPUP)S=-1.
      X(I)=HI-S*RI*SIN(PIO180*THI)
      Z(I)=KI+S*RI*COS(PIO180*THI)
      GO TO 4130
C ICASE=14
 4200 B(I)=ZI-XI*TAN(PIO180*THI)
      GO TO 4040
C ICASE=15
 4300 Z(I)=XI*TAN(PIO180*THI)+BI
      GO TO 4040
C ICASE=16
 4400 X(I)=(ZI-BI)/TAN(PIO180*THI)
      GO TO 4040
C ICASE=17
 4410 IF(ABS(Z(I)-X(I)*TAN(PIO180*TH(I))-B(I)).LT.ERRMAX)GO TO 4150
      JSTOP=5
      WRITE(IW,4050)JSTOP,I,ERRMAX
      GO TO 4150
C ICASE=18
```

Table 8. Continued

```
 4420 IF(ABS((X(I)-H(I))**2+(Z(I)-K(I))**2-R(I)**2).LT.ERRMAX)GO TO 4150
      KSTOP=7
      WRITE(IW,4140)KSTOP,I,ERRMAX
      GO TO 4150
C ICASE=19
 4430 S=0.
      IF(DIR(I).EQ.UPFO)S=+1.
      IF(DIR(I).EQ.UPAF)S=+1.
      IF(DIR(I).EQ.UPTO)S=+1.
      IF(DIR(I).EQ.UPBO)S=+1.
      IF(DIR(I).EQ.DNFO)S=-1.
      IF(DIR(I).EQ.DNAF)S=-1.
      IF(DIR(I).EQ.DNTO)S=-1.
      IF(DIR(I).EQ.DNBO)S=-1.
      IF(S.NE.0.)GO TO 4440
      WRITE(IW,4431)I
 4431 FORMAT(' COND CODE 4431: DIR(I) EQUAL TO ILLEGAL VALUE FOR CASE 19
     1 FOR SEGMENT',I3)
      WRITE(IW,2035)
      STOP 4431
 4440 IF(I.NE.NSEG)GO TO 4450
      WRITE(IW,4441)I
 4441 FORMAT(' COND CODE 4441: ICASE=19 MAY NOT BE USED FOR LAST SEGMENT
     1',I3)
      WRITE(IW,2035)
      STOP 4441
 4450 H(I)=XJ-S*RI  *SIN(PIO180*THJ)
      K(I)=ZJ+S*RI  *COS(PIO180*THJ)
      IF(DIR(I).EQ.UPFO)DIR(I)=FORE
      IF(DIR(I).EQ.UPAF)DIR(I)=AFTE
      IF(DIR(I).EQ.UPTO)DIR(I)=TOPS
      IF(DIR(I).EQ.UPBO)DIR(I)=BOTS
      IF(DIR(I).EQ.DNFO)DIR(I)=FORE
      IF(DIR(I).EQ.DNAF)DIR(I)=AFTE
      IF(DIR(I).EQ.DNTO)DIR(I)=TOPS
      IF(DIR(I).EQ.DNBO)DIR(I)=BOTS
      ICASE(I)=5
      GO TO 4130
C ICASE=20
 4460 IM1=I-1
      A1=SQRT((XI-H(IM1))**2+(ZI-K(IM1))**2)
      A2=ARCOS(R(IM1)/A1)/PIO180
      A3=ATAN((ZI-K(IM1))/(XI-H(IM1)))/PIO180
      S=0.
      IF(DIR(I).EQ.TOPS)S=+1.
      IF(DIR(I).EQ.BOTS)S=-1.
      IF(S.NE.0.)GO TO 4470
      WRITE(IW,4461)I
 4461 FORMAT(' COND CODE 4461: DIR MUST BE EITHER TOPS OR BOTS FOR SEGME
     1NT',I3)
      WRITE(IW,2035)
      STOP 4461
 4470 IF(I.NE.1)GO TO 4480
      WRITE(IW,4471)
 4471 FORMAT(' COND CODE 4471: ICASE=20 MAY NOT BE USED FOR FIRST SEGMEN
     1T')
```

Table 8. Continued

```
      WRITE(IW,2035)
      STOP 4471
 4480 TH(I)=A3-S*(90.-A2)
      M(I)=TAN(PIO180*TH(I))
      B(I)=ZI-M(I)*XI
      GO TO 4040
C ICASE=21
 4490 IF(I.NE.1)GO TO 4500
      WRITE(IW,4491)
 4491 FORMAT(' COND CODE 4491: ICASE=21 MAY NOT BE USED WITH FIRST SEGEM
     1ENT')
      WRITE(IW,2035)
      STOP 4491
 4500 IF(CURVE(I-1).EQ.OGIV)GO TO 4540
C LINE-OGIVE
      IM1=I-1
      MJ =TAN(PIO180*TH(IM1))
      A1=MJ*X(I)+B(IM1)
      S=+1.
      IF(A1.GT.Z(I))S=-1.
      BJ=B(IM1)+S*R(I)/COS(PIU180*TH(IM1))
      A1=1.+MJ**2
      A2=2.*((BJ-Z(I))*MJ-X(I))
      A3=X(I)**2+(BJ-Z(I))**2-R(I)**2
      A4=A2/2./A1
      A5=A3/A1
      A5=A4**2-A5
      IF(A5.GE.0.)GO TO 4515
      IF(ABS(A5).GE.ERRMAX)GO TO 4510
      WRITE(IW,4501)I
 4501 FORMAT(' COND CODE 4501: SEGMENT',I3,'  LOCUS OF CIRCLE CENTERS DO
     1ES NOT INTERSECT')
      WRITE(IW,2035)
      STOP 4501
 4510 A5=0.
 4515 A5=SQRT(A5)
      A1=-A4+A5
      A2=-A4-A5
      A3=A1*MJ+BJ
      A4=A2*MJ+BJ
      IF(A1.GT.A2)GO TO 4520
      H(I)=A1
      K(I)=A3
      GO TO 4530
 4520 H(I)=A2
      K(I)=A4
 4530 TH(I)=-ATAN((X(I)-H(I))/(Z(I)-K(I)))/PIO180
      GO TO 4130
C OGIVE-OGIVE
 4540 IM1=I-1
      IF(DIR(IM1).EQ.OPDN.OR.UIR(IM1).EQ.OPUP)GO TO 4550
 4541 WRITE(IW,4542)IM1
 4542 FORMAT(' COND CODE 4542: SEGMENT',I3,' DIR NOT EQUAL TO OPUP OR OP
     1DN')
      WRITE(IW,2035)
      STOP 4542
```

Table 8. Continued

```
4550 IF(DIR(I).EQ.OPUP.OR.DIR(I).EQ.OPDN)GO TO 4560
     IM1=I
     GO TO 4541
4560 S=+1.
     IF(DIR(IM1).EQ.DIR(I))S=-1.
     IF(Z(I).EQ.K(IM1))GO TO 4630
     MJ=-(X(I)-H(IM1))/(Z(I)-K(IM1))
     BJ=X(I)**2+Z(I)**2-H(IM1)**2-K(IM1)**2+R(IM1)*(R(IM1)+2.*S*R(I))
     BJ=BJ*.5/(Z(I)-K(IM1))
     A1=1+MJ**2
     A2=2.*((BJ-Z(I))*MJ-X(I))
     A3=X(I)**2+(BJ-Z(I))**2-R(I)**2
     A4=.5*A2/A1
     A5=A3/A1
     A5=A4**2-A5
     IF(A5.GE.0.)GO TO 4580
     IF(ABS(A5).GE.ERRMAX)GO TO 4570
     WRITE(IW,4561)I
4561 FORMAT(' COND CODE 4561: SEGMENT',I3,'  LOCUS OF CIRCLE CENTERS DO
    1ES NOT INTERSECT')
     WRITE(IW,2035)
     STOP 4561
4570 A5=0.
4580 A5=SQRT(A5)
     A1=-A4+A5
     A2=-A4-A5
     A3=MJ*A1+BJ
     A4=MJ*A2+BJ
4585 IF(DIR(IM1).EQ.DIR(I))GO TO 4622
     IF(DIR(IM1).EQ.OPDN.AND.DIR(I).EQ.OPUP)GO TO 4590
     IF(DIR(IM1).EQ.OPUP.AND.DIR(I).EQ.OPDN)GO TO 4620
     STOP 4581
4590 IF(A4.GT.A3)GO TO 4610
4600 H(I)=A1
     K(I)=A3
     GO TO 4660
4610 H(I)=A2
     K(I)=A4
     GO TO 4660
4620 IF(A4.GT.A3)GO TO 4600
     GO TO 4610
4630 A1 =(X(I)**2+Z(I)**2-(H(IM1)**2+K(IM1)**2)+R(IM1)*(R(IM1)+2.*
    1 S*R(I)))*.5/(X(I)-H(IM1))
     A2=R(I)**2-(A1 -X(I))**2
     IF(A2.GT.0.)GO TO 4650
     IF(ABS(A2).GE.ERRMAX)GO TO 4640
     WRITE(IW,4561)I
     STOP 4562
4640 A2=0.
4650 A2=SQRT(A2)
     A3=Z(I)+A2
     A4=Z(I)-A2
     A2=A1
     GO TO 4585
4622 IF(DIR(IM1).EQ.OPDN.AND.DIR(I).EQ.OPDN)GO TO 4620
     IF(DIR(IM1).EQ.OPUP.AND.DIR(I).EQ.OPUP)GO TO 4590
```

Table 8. Continued

```
      STOP 4623
 4660 TH(I)=-ATAN((X(I)-H(I))/(Z(I)-K(I)))/PIO180
      GO TO 4130
 4990 CONTINUE
      IF(JSTOP.NE.0)WRITE(IW,2035)
      IF(KSTOP.NE.0)WRITE(IW,2035)
      IF(ISTOP.NE.0)WRITE(IW,2035)
      IF(JSTOP.NE.0)STOP 4050
      IF(KSTOP.NE.0)STOP 4140
      IF(ISTOP.NE.0)STOP 4005
      RETURN
 2035 FORMAT('0EXECUTION HALTED')
      END
C***********************************************************************
      SUBROUTINE INTR
C--------------------------------------------
C COMPUTE SEGMENT INTERSECTIONS FOR ICASE=5,9
C--------------------------------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),R-OSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NN1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
 5000 CONTINUE
      DO 5320 I=1,NSEG
      IP1=I+1
      IF(ICASE(I).NE.5.AND.ICASE(I).NE.9)GO TO 5320
      IF(I.NE.NSEG)GO TO 5020
      WRITE(IW,5010)
 5010 FORMAT(' COND CODE 5010: EQUATION IS INSUFFICIENT FOR LAST SEGMENT
     1')
      WRITE(IW,2035)
      STOP 5010
 5020 CONTINUE
      IF(ICASE(I).EQ.5.AND.CURVE(IP1).EQ.OGIV)GO TO 5040
      IF(ICASE(I).EQ.5.AND.CURVE(IP1).EQ.LINE)GO TO 5250
      IF(ICASE(I).EQ.9.AND.CURVE(IP1).EQ.OGIV)GO TO 5260
      IF(ICASE(I).EQ.9.AND.CURVE(IP1).EQ.LINE)GO TO 5240
      WRITE(IW,5025)
 5025 FORMAT(' COND CODE 5030: OOPS: PROGRAMMER ERROR.')
      WRITE(IW,2035)
 5030 STOP 5030
C OGIVE-OGIVE INTERSECTION
 5040 CONTINUE
```

Table 8. Continued

```
      THIJ=ATAN2(K(IP1)-K(I),H(IP1)-H(I))/PIO180
      SIJ=SQRT((H(IP1)-H(I))**2+(K(IP1)-K(I))**2 )
      AI=      (R(I)**2+SIJ**2-R(IP1)**2)/(2.*R(I)*SIJ)
      IF(ABS(AI).GT.1.)AI=SIGN(1.,AI)
      AI=ARCOS(AI)/PIO180
      XI=H(I)+R(I)*COS((THIJ+AI)*PIO180)
      X(I)=H(I)+R(I)*COS((THIJ-AI)*PIO180)
      IF(ABS(X(I)-XI) .LE.ERRMAX)GO TO 5090
      IF(I.LT.NSEG.AND.SLOPE(I+1).EQ.T)GO TO 5070
      IF(DIR(I).EQ.FORE)GO TO 5060
      IF(DIR(I).EQ.AFTE)GO TO 5080
      WRITE(IW,5050)I
 5050 FORMAT(' COND CODE 5050: DIR MUST BE EITHER ''FORE'' OR ''AFTE'' F
     1OR SEGMENT ',I2)
      WRITE(IW,2035)
      STOP 5050
 5060 IF(XI.GE.X(I))GO TO 5070
 5065 X(I)=XI
      Z(I)=K(I)+R(I)*SIN(PIO180*(THIJ+AI))
      GO TO 5230
 5070 Z(I)=K(I)+R(I)*SIN(PIO180*(THIJ-AI))
      GO TO 5230
 5080 IF(XI.GT.X(I))GO TO 5065
      GO TO 5070
 5090 ZI=K(I)+R(I)*SIN(PIO180*(THIJ+AI))
      Z(I)=K(I)+R(I)*SIN(PIO180*(THIJ-AI))
      IF(ABS(Z(I)-ZI).LE.ERRMAX)GO TO 5230
      IF(DIR(I).EQ.TOPS)GO TO 5200
      IF(DIR(I).EQ.BOTS)GO TO 5220
      WRITE(IW,5100)I
 5100 FORMAT(' COND CODE 5100: DIR MUST BE EITHER ''TOPS'' OR ''BOTS'' F
     1OR SEGMENT ',I2)
      WRITE(IW,2035)
      STOP 5100
 5200 IF(ZI.GT.Z(I))GO TO 5210
 5205 X(I)=H(I)+R(I)*COS(PIO180*(THIJ-AI))
      GO TO 5230
 5210 Z(I)=ZI
      X(I)=H(I)+R(I)*COS(PIO180*(THIJ+AI))
      GO TO 5230
 5220 IF(ZI.GT.Z(I))GO TO 5205
      GO TO 5210
 5230 TH(I)=-ATAN((X(I)-H(I))/(Z(I)-K(I)))/PIO180
      GO TO 5320
C LINE-LINE INTERSECTION
 5240 M(I  )=TAN(PIO180*TH(I  ))
      M(IP1)=TAN(PIO180*TH(IP1))
      X(I)=-(B(IP1)-B(I))/(M(IP1)-M(I))
      Z(I)=(M(I)*B(IP1)-M(IP1)*B(I))/(M(I)-M(IP1))
      GO TO 5320
C OGIVE-LINE INTERSECTION
 5250 CONTINUE
      M(IP1)=TAN(PIO180*TH(IP1))
      MI=M(IP1)
      BI=B(IP1)
      HJ=H(I)
```

Table 8. Continued

```
      KJ=K(I)
      RJ=R(I)
      GO TO 5270
C LINE-OGIVE INTERSECTION
 5260 CONTINUE
      M(I)=TAN(PIO180*TH(I))
      MI=M(I)
      BI=B(I)
      HJ=H(IP1)
      KJ=K(IP1)
      RJ=R(IP1)
      GO TO 5270
 5270 A1=1.+MI**2
      A2=MI*(BI-KJ)-HJ
      A3=HJ**2+(BI-KJ)**2-RJ**2
      A4=A2**2-A1*A3
      IF(A4.GT.0.)GO TO 5273
      IF(ABS(A4).LT.ERRMAX)GO TO 5272
      WRITE(IW,5271)I
 5271 FORMAT(' COND CODE 5271: SQRT(<0) FOR SEGMENT',I3,' NO INTERSECTIO
     1N.')
      WRITE(IW,2035)
      STOP 5271
 5272 A4=0.
 5273 A4=SQRT(A4)
      XI=(-A2+A4)/A1
      X(I)=(-A2-A4)/A1
      IF(ABS(X(I)-XI).LE.ERRMAX)GO TO 5300
      IF(I.LT.NSEG.AND.SLOPE(I+1).EQ.T)GO TO 5300
      IF(DIR(I).EQ.FORE)GO TO 5290
      IF(DIR(I).EQ.AFTE)GO TO 5310
      WRITE(IW,5280)I
 5280 FORMAT(' CONB CODE 5280: DIR MUST BE EITHER ''FORE'' OR ''AFTE'' F
     1OR SEGMENT',I2)
      WRITE(IW,2035)
      STOP 5280
 5290 IF(XI.GT.X(I))GO TO.5300
 5295 X(I)=XI
 5300 Z(I)=MI*X(I)+BI
      GO TO 5320
 5310 IF(XI.GT.X(I))GO TO 5295
      GO TO 5300
 5320 CONTINUE
      RETURN
 2035 FORMAT('0EXECUTION HALTED')
      END
C***********************************************************************
      SUBROUTINE QARY
C-----------------------------------
C SET UP FINAL ARRAYS FOR QUICK MODEL
C-----------------------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),R OSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
```

Table 8. Continued

```
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
C TQ(I,1) IS AT BEGINNING OF SEGMENT I
C TQ(I,2) IS AT END SEGMENT I
C XQ(I) AND ZQ(I) ARE AT BEGINNING OF SEGMENT I
 6000 CONTINUE
      XQ(1)=0.
      ZQ(1)=0.
      DO 6010 I=1,NSEG
      IF(CURVE(I).EQ.KRV(2))CURVE(I)=ELLX
      XQ(I+1)=X(I)
      ZQ(I+1)=Z(I)
 6010 CONTINUE
      DO 6030 I=1,NSEG
      IP1=I+1
      IF(CURVE(I).NE.LINE)GO TO 6020
      TQ(I,1)=ATAN((ZQ(IP1)-ZQ(I))/(XQ(IP1)-XQ(I)))/PIO180
      TQ(I,2)=TQ(I,1)
      GO TO 6030
 6020 IF(ZQ(I).EQ.K(I))GO TO 6025
      TQ(I,1)=-ATAN((XQ(I)-H(I))/(ZQ(I)-K(I)))/PIO180
      GO TO 6024
 6025 TQ(I,1)=90.
 6024 TQ(I,2)=-ATAN((XQ(IP1)-H(I))/(ZQ(IP1)-K(I)))/PIO180
 6030 CONTINUE
      NSEGP1=NSEG+1
C SET ALL RIGHT ANGLES (90 DEG) TO "IBM RIGHT ANGLES" (89.999 DEG)
      DO 6040 I=1,NSEGP1
      DO 6040 J=1,2
      IF(TQ(I,J).GT.RINF)GO TO 6040
      IF( ABS(TQ(I,J)).LE.89.999)GO TO 6040
      TQ(I,J)=SIGN(89.999,TQ(I,J))
 6040 CONTINUE
      RETURN
      END
C******************************************************************
      SUBROUTINE PRNT
C-------------
C PRINT RESULTS
C-------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
```

Table 8. Continued

```
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
 7000 CONTINUE
      WRITE(IW,7005)
 7005 FORMAT('0SEGMENT END POINT COORDINATES AND SLOPES:')
      WRITE(IW,7010)
 7010 FORMAT('0',9 X,'X',13X,'Z',12X,'TA',12X,'TF'/' ',58('-'))
      DO 7020 I=1,11
      WRITE(IW,90008)I,XQ(I),ZQ(I),TQ(I,2),TQ(I,1)
 7020 CONTINUE
      WRITE(IW,7025)
 7025 FORMAT('0SEGMENT EQUATIONS:')
      WRITE(IW,7030)
 7030 FORMAT(      '0', 9X,'X',13X,'H',13X,'K',13X,'R',13X,'M',13X,'T',
     2 13X,'B',/' ',100('-'))
      DO 7040 I=1,10
      WRITE(IW,90008)I,X(I),H(I),K(I),R(I),M(I),TH(I),B(I)
 7040 CONTINUE
      RETURN
90008 FORMAT(' ',I2,1P9E14.6)
      END
C***********************************************************************
      SUBROUTINE ROFL
C-----------------------------
C INSERT ROUNDS AND FILLETS
C-----------------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
 6100 CONTINUE
      ISKIP=0
      DO 6800 I=1,9
      IP1=I+1
```

Table 8. Continued

```
      IF(RHO(I).GT.RINF)GO TO 6800
      ISKIP=1
      IF(CURVE(I).EQ.LINE.AND.CURVE(IP1).EQ.LINE)GO TO 6110
      IF(CURVE(I).EQ.LINE.AND.CURVE(IP1).EQ.ELLX)GO TO 6200
      IF(CURVE(I).EQ.ELLX.AND.CURVE(IP1).EQ.LINE)GO TO 6300
      IF(CURVE(I).EQ.ELLX.AND.CURVE(IP1).EQ.ELLX)GO TO 6600
      WRITE(IW,6101)I
 6101 FORMAT(' COND CODE 6101: PROGRAM OR USER ERROR, SEGMENT',I3)
      WRITE(IW,2035)
      STOP 6101
C LINE-LINE FILLETTING/ROUNDING
 6110 CONTINUE
      IF(ABS(TQ(I,2)-TQ(IP1,1)).GT.ERRMAX)GO TO 6120
      WRITE(IW,6111)I,IP1
 6111 FORMAT(' COND CODE 6111: FILLETS/ROUNDS MAY NOT BE INSERTED BETWEE
     1NLINE SEGMENTS WITH SAME SLOPE,SEGMENTS',2I4)
      WRITE(IW,2035)
      STOP 6111
 6120 S=1.
      IF(TQ(IP1,1).GT.TQ(I,2))S=-1.
      BI=B(I)+S*RHO(I)/COS(PIO180*TQ(I,2))
      BJ=B(IP1)-S*RHO(I)/COS(PIO180*TQ(IP1,1))
      M(I)=TAN(PIO180*TH(I))
      M(IP1)=TAN(PIO180*TH(IP1))
      ETA(I)=(M(IP1)*BI-M(I)*BJ)/(M(IP1)-M(I))
      KSI(I)=-(BJ-BI)/(M(IP1)-M(I))
      X1(I)=KSI(I)-S*RHO(I)*SIN(PIO180*TH(I   ))
      X2(I)=KSI(I)-S*RHO(I)*SIN(PIO180*TH(IP1))
      Z1(I)=ETA(I)+S*RHO(I)*COS(PIO180*TH(I   ))
      Z2(I)=ETA(I)+S*RHO(I)*COS(PIO180*TH(IP1))
      T1(I)=TH(I)
      T2(I)=TH(IP1)
      GO TO 6800
C LINE-OGIVE FILLETTING/ROUNDING
 6200 CONTINUE
C OGIVE-LINE FILLETTING/ROUNDING
 6300 CONTINUE
 6201 S1=0.
      S2=0.
      S3=0.
      I1=I
      IF(CURVE(IP1).EQ.ELLX)I1=IP1
      I2=I
      IF(I1.EQ.I)I2=IP1
      IF(TQ(I,2)-TQ(IP1,1))6210,6205,6230
 6205 WRITE(IW,6206)I,IP1
 6206 FORMAT(' COND CODE 6206: FILLETS/ROUNDS MAY NOT BE INSERTED BETWEE
     1N SEGMENTS WITH SAME SLOPE, SEGMENTS',2I3)
      WRITE(IW,2035)
      STOP 6206
C TM(IP1)>TH(I)
 6210 IF(K(I1)-ZQ(IP1))6220,6215,6225
 6215 WRITE(IW,6216)I,IP1
 6216 FORMAT(' COND CODE 6216: K = Z NOT A PROGRAMMED OPTION, SEGMENTS',
     1 2I3)
      WRITE(IW,2035)
```

Table 8. Continued

```
      STOP 6216
 6220 S1=1.
      S2=1.
      S3=1.
      GO TO 6250
 6225 S1=1.
      S2=-1.
      S3=1.
      GO TO 6250
 6230 IF(K(I1)-ZQ(IP1))6240,6235,6245
C TH(IP1)<TH(I)
 6240 S1=-1.
      S2=-1.
      S3=-1.
      GO TO 6250
 6235 WRITE(IW,6236)I,IP1
 6236 FORMAT(' COND CODE 6236: K = Z NOT A PROGRAMMED OPTION, SEGMENTS',
     1 2I3)
      WRITE(IW,2035)
      STOP 6236
 6245 S1=-1.
      S2=1.
      S3=-1.
      GO TO 6250
 6250 CONTINUE
      BI=B(I2)+S1*RHO(I)/COS(PIO180*TH(I2))
      A1=1.+M(I2)**2
      A2=2.*((BI-K(I1))*M(I2)-H(I1))
      A3=H(I1)**2+(BI-K(I1))**2-(R(I1)+S2*RHO(I))**2
      A4=0.5*A2/A1
      A5=A3/A1
      A5=A4**2-A5
      IF(A5.GE.0.)GO TO 6260
      IF(ABS(A5).GT.ERRMAX)GO TO 6254
      A5=0.
      GO TO 6260
 6254 WRITE(IW,6255)I,IP1
 6255 FORMAT(' COND CODE 6255:  REAL SOLUTION TO FILLET/ROUND PROBLEM DO
     1ES NOT EXIST.  POSSIBLE PROGRAM ERROR. SEGMENTS ',2I2)
      WRITE(IW,2035)
      STOP 6255
 6260 A5=SQRT(A5)
      A1=-A4-A5
      A2=M(I2)*A1+BI
      A6=H(I1)+R(I1)*(A1-H(I1))/SQRT((A1-H(I1))**2+(A2-K(I1))**2)
      A7=(M(I2)*(A2-B(I2))+A1)/(1.+M(I2)**2)
      IF(X(I).GE.AMIN1(A6,A7).AND.X(I).LE.AMAX1(A6,A7))GO TO 6265
      A1=-A4+A5
      A2=M(I2)*A1+BI
      A6=H(I1)+R(I1)*(A1-H(I1))/SQRT((A1-H(I1))**2+(A2-K(I1))**2)
      A7=(M(I2)*(A2-B(I2))+A1)/(1.+M(I2)**2)
      IF(X(I).GE.AMIN1(A6,A7).AND.X(I).LE.AMAX1(A6,A7))GO TO 6265
      WRITE(IW,6261)I
 6261 FORMAT(' COND CODE 6261: PROGRAM UNABLE TO CHOOSE BETWEEN TWO FILL
     1ET/ROUND SOLUTIONS, SEGMENT',I3)
      WRITE(IW,2035)
```

Table 8. Continued

```
      STOP 6261
 6265 KSI(I)=A1
      ETA(I)=A2
      IF(CURVE(I).EQ.ELLX)GO TO 6270
      X1(I)=A7
      X2(I)=A6
      Z2(I)=K(I1)+R(I1)*(ETA(I)-K(I1))/SQRT((A1-H(I1))**2+(A2-K(I1))**2)
      Z1(I)=M(I2)*X1(I)+B(I2)
      T1(I)=TH(I)
      T2(I)=-ATAN((X2(I)-H(I1))/(Z2(I)-K(I1)))/PIO180
      GO TO 6800
 6270 X1(I)=A6
      X2(I)=A7
      Z2(I)=M(I2)*X2(I)+B(I2)
      Z1(I)=K(I1)+R(I1)*(ETA(I)-K(I1))/SQRT((A1-H(I1))**2+(A2-K(I1))**2)
      T1(I)=-ATAN((X1(I)-H(I1))/(Z1(I)-K(I1)))/PIO180
      T2(I)=TH(IP1)
      GO TO 6800
 6600 CONTINUE
C OGIVE-OGIVE FILLETTING ROUNDING
      A1=TQ(IP1,1)-TQ(I,2)
      A2=Z(I)-K(I)
      A3=Z(I)-K(IP1)
      IF(A1.GT.ZERO.AND.A2.GT.ZERO.AND.A3.GT.ZERO)GO TO 6610
      IF(A1.LT.ZERO.AND.A2.GT.ZERO.AND.A3.GT.ZERO)GO TO 6615
      IF(A1.LT.ZERO.AND.A2.LT.ZERO.AND.A3.LT.ZERO)GO TO 6610
      IF(A1.GT.ZERO.AND.A2.LT.ZERO.AND.A3.LT.ZERO)GO TO 6615
      IF(A1.LT.ZERO.AND.A2.GT.ZERO.AND.A3.LT.ZERO)GO TO 6620
      IF(A1.GT.ZERO.AND.A2.GT.ZERO.AND.A3.LT.ZERO)GO TO 6625
      IF(A1.GT.ZERO.AND.A2.LT.ZERO.AND.A3.GT.ZERO)GO TO 6620
      IF(A1.LT.ZERO.AND.A2.LT.ZERO.AND.A3.GT.ZERO)GO TO 6625
      WRITE(IW,6601)I,IP1,A1,A2,A3
 6601 FORMAT(' COND CODE 6601: SEGMENTS',2I3,', A1,A2,A3 MAY NOT BE ZERO
     1.  A1,A2,A3=',1P3E15.7)
      WRITE(IW,2035)
      STOP 6601
 6610 S1=1.
      S2=1.
      GO TO 6630
 6615 S1=-1.
      S2=-1.
      GO TO 6630
 6620 S1=-1.
      S2=1.
      GO TO 6630
 6625 S1=1.
      S2=-1.
      GO TO 6630
 6630 CONTINUE
      RI=R(I)+S1*RHO(I)
      RJ=R(IP1)+S2*RHO(I)
      IF(K(I).EQ.K(IP1))GO TO 6645
      MI=-(H(IP1)-H(I))/(K(IP1)-K(I))
      BI=(H(IP1)**2-H(I)**2+K(IP1)**2-K(I)**2-RJ**2+RI**2)/2./(K(IP1)-
     1 K(I))
      A1=1.+MI**2
```

Table 8. Continued

```
      A2=2.*((BI-K(I))*MI-H(I))
      A3=H(I)**2+(BI-K(I))**2-RI**2
      A4=.5*A2/A1
      A5=A3/A1
      A5=A4**2-A5
      IF(A5.GE.0.)GO TO 6635
      IF(ABS(A5).LE.ERRMAX)GO TO 6634
      WRITE(IW,6631)I,IP1
 6631 FORMAT(' COND CODE 6631: REAL SOLUTION TO FILLET/ROUND PROBLEM DOE
     1S NOT EXIST. POSSIBLE PROGRAM ERROR. SEGMENTS',2I3)
      WRITE(IW,2035)
      STOP 6631
 6634 A5=0.
 6635 A5=SQRT(A5)
      A1=-A4+A5
      A2=-A4-A5
      A3=MI*A1+BI
      A4=MI*A2+BI
 6636 A5=SQRT((X(I)-A1)**2+(Z(I)-A3)**2)
      A6=SQRT((X(I)-A2)**2+(Z(I)-A4)**2)
      IF(A6.LT.A5)GO TO 6640
      KSI(I)=A1
      ETA(I)=A3
      GO TO 6650
 6640 KSI(I)=A2
      ETA(I)=A4
      GO TO 6650
 6645 A1=(H(IP1)**2-H(I)**2-RJ**2+RI**2)/2./(H(IP1)-H(I))
      A2=A1
      A4=RI**2-(A1-H(I))**2
      IF(A4.GE.0.)GO TO 6648
      IF(ABS(A4).GT.ERRMAX)GO TO 6646
      A4=0.
 6648 A4=SQRT(A4)
      A3=K(I)-A4
      A4=K(I)+A4
      GO TO 6636
 6646 WRITE(IW,6647)I,IP1
 6647 FORMAT(' COND CODE 6647: FOR SEGMENTS',2I3,' K(I) = K(IP1): REAL S
     1OLUTION TO FILLET/ROUND PROBLEM DOES NOT EXIST. POSSIBLE PROGRAM E
     2RROR.')
      WRITE(IW,2035)
      STOP 6647
 6650 DO 6660 J=I,IP1
      A1=SQRT((KSI(I)-H(J))**2+(ETA(I)-K(J))**2)
      A2=H(J)+R(J)*(KSI(I)-H(J))/A1
      A3=K(J)+R(J)*(ETA(I)-K(J))/A1
      A4=-ATAN((A2-H(J))/(A3-K(J)))/PIO180
      IF(J.EQ.IP1)GO TO 6655
      X1(I)=A2
      Z1(I)=A3
      T1(I)=A4
      GO TO 6660
 6655 X2(I)=A2
      Z2(I)=A3
      T2(I)=A4
```

Table 8. Continued

```
 6660 CONTINUE
 6800 CONTINUE
      IF(ISKIP.EQ.0)RETURN
C PRINT RESULTS OF FILLET/ROUND COMPUTATION
      WRITE(IW,6810)
 6810 FORMAT('0FILLET/ROUND EQUATIONS AND INTERSECTIONS:'/
     1'0', 7X,' X ',10X,'KSI',10X,'ETA',10X,'RHO',10X,' X1',10X,' Z1',
     2    10X,' T1',10X,' X2',10X,' Z2',10X,' T2'/
     3' ',132('-'))
      WRITE(IW,6815)(I,X(I),KSI(I),ETA(I),RHO(I),X1(I),Z1(I),T1(I),
     1 X2(I),Z2(I),T2(I),I=1,10)
 6815 FORMAT(10(' ',I2,1P10E13.5/))
C INSERT FILLET/ROUND DATA INTO QUICK ARRAYS
 6900 CONTINUE
      J=0
      DO 6920 I=1,10
      J=J+1
      IF(RHO(I).GT.RINF)GO TO 6920
      J=J+1
      NSEG=NSEG+1
      NSEGP1=NSEG+1
      IF(NSEG.LE.10)GO TO 6905
      WRITE(IW,6901)I
 6901 FORMAT(' COND CODE 6901: INSERTION OF ROUND',I3,' EXCEEDS DIMENSIO
     1NS (10) OF QUICK')
      WRITE(IW,2035)
      STOP 6901
 6905 L=11
 6910 L=L-1
      LM1=L-1
      IF(L.LE.J)GO TO 6915
      XQ(L)=XQ(LM1)
      ZQ(L)=ZQ(LM1)
      TQ(L,1)=TQ(LM1,1)
      TQ(L,2)=TQ(LM1,2)
      CURVE(L)=CURVE(LM1)
      GO TO 6910
 6915 XQ(J)=X1(I)
      ZQ(J)=Z1(I)
      TQ(J,1)=T1(I)
      TQ(J,2)=T2(I)
      TQ(J-1,2)=T1(I)
      CURVE(J+1)=CURVE(J)
      XQ(J+1)=X2(I)
      ZQ(J+1)=Z2(I)
      TQ(J+1,1)=T2(I)
      CURVE(J)=ELLX
 6920 CONTINUE
C PRINT RESULTS
      WRITE(IW,6930)
 6930 FORMAT('0FINAL QUICK ARRAY:')
      WRITE(IW,7010)
      DO 6940 I=1,10
 6940 WRITE(IW,90008)I,XQ(I),ZQ(I),TQ(I,2),TQ(I,1)
      RETURN
 2035 FORMAT('0EXECUTION HALTED')
```

Table 8. Continued

```
 7010 FORMAT('0',9 X,'X',13X,'Z',12X,'TA',12X,'TF'/' ',58('-'))
90008 FORMAT(' ',I2,1P9E14.6)
      END
C**********************************************************************
      SUBROUTINE QUCK
C------------------
C SET UP QUICK MODEL
C------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
 8000 CONTINUE
      IF(IF.NE.IP)REWIND IF
C CROSS SECTION MODELS
      WRITE(IF,90001)IDEN
      IF(ISL.NE.0)GO TO 8001
      WRITE(IF,90002)
      WRITE(IF,90003)
      GO TO 8002
 8001 WRITE(IF,21700)
 8002 CONTINUE
      I1=1
      WRITE(IF,90004)I1,I1,XQ(1),XQ(NSEGP1)
C BODY LINE MODELS
      WRITE(IF,90005)YUCL
      WRITE(IF,90006)I1,KRV(1),KV(1)
      WRITE(IF,90007)ZERO,ZERO,XQ(NSEGP1),ZERO,ZERO,ZERO
      WRITE(IF,90009)
      DO 8030 J=1,2
      WRITE(IF,90005)CL(J)
      DO 8010 I=1,NSEG
      IP1=I+1
      I1=1
      IF(CURVE(I).EQ.ELLX)I1=2
      WRITE(IF,90006)I,CURVE(I),KV(I1)
      GO TO (8006,8005),I1
 8005 CONTINUE
      WRITE(IF,90007)XQ(I),ZQ(I),XQ(IP1),ZQ(IP1),TQ(I,1),TQ(I,2)
      GO TO 8010
 8006 WRITE(IF,90014)XQ(I),ZQ(I),XQ(IP1),ZQ(IP1)
 8010 CONTINUE
      WRITE(IF,90009)
```

Table 8. Continued

```
      DO 8020 I=1,NSEGP1
      ZQ(I)=-ZQ(I)
      TQ(I,1)=-TQ(I,1)
      TQ(I,2)=-TQ(I,2)
 8020 CONTINUE
 8030 CONTINUE
      WRITE(IF,90010)
      IF(ISL.EQ.0)GO TO 8033
      DO 8032  L=1,3
      WRITE(IF,90005)SL(L)
      I1=NS(L)
      DO 8031  I=1,I1
      I2=1
      IF(CRVSL(I,L).EQ.ELLX)I2=2
      WRITE(IF,90006)I,CRVSL(I,L),KV(I2)
      WRITE(IF,90007)V(I,1,L),V(I,2,L),V(I+1,1,L),V(I+1,2,L),V(I,3,L),
     1 V(I+1,3,L)
 8031 CONTINUE
      WRITE(IF,90009)
 8032 CONTINUE
      WRITE(IF,21900)
 8033 CONTINUE
      WRITE(IF,90011)
      IF(IPRINT.EQ.0)GO TO 8050
      DX=.00001
      NPTS=5
C SET UP CARDS TO EXERCISE MODEL
      I1=1
      I2=2
      DO 8040 I=1,NSEG
      A1=XQ(I)+DX
      A2=XQ(I+1)-DX
      A3=(A2-A1)/(NPTS-1)
      A2=XQ(I+1)+DX
      WRITE(IF,90012)I1,I2,A1,A2,A3
 8040 CONTINUE
      IF(ISL.EQ.0)GO TO 8050
      I1=2
      I2=2
      I4=NP(1)
      A3=2.
      A4=-90.
      A5=90.
      A6=5.
      DO 8044 L=1,3
      I3=NP(L)
      DO 8043 I=2,I3
      IF(L.EQ.1)GO TO 8042
      IF(I.EQ.J)GO TO 8043
      DO 8041 J=1,I4
      IF(ABS(V(J,1,1)-V(I,1,L)).LE.ERRMAX)GO TO 8043
 8041 CONTINUE
 8042 A1=V(I,1,L)
      A2=A1+1.
      WRITE(IF,90012)I1,I2,A1,A2,A3,A4,A5,A6
 8043 CONTINUE
```

Table 8. Continued

```
 8044 CONTINUE
 8050 WRITE(IF,90013)
      RETURN
C-------------------
C QUICK INPUT FORMATS
C-------------------
90001 FORMAT(80A1)
90002 FORMAT(' 1'/' 1 2')
90003 FORMAT('LELL     1ELLI  PIEC    BLCL       SID       LSCP'/
     1       'UELL     2ELLI  PIEC     SID      TUCL       USCP'/
     2       ' 1        MAPAX')
90004 FORMAT(2I2,6X,2F10.5)
90005 FORMAT(A4)
90006 FORMAT(I2,1X,A4,'   PIEC  ',A4)
90007 FORMAT(3F10.5,2(F9.5,'A'),F10.5)
90008 FORMAT(' ',I2,1P9E14.6)
90009 FORMAT('-1')
90010 FORMAT('YLCL      YUCL'/
     1       'YMAPAX    YUCL'/
     2       'ZMAPAX    YUCL'/
     3       'ZSID      YUCL'/
     4       'YUSCP     ZUCL'/
     5       'ZUSCP     ZUCL'/
     6       'YLSCP     ZUCL'/
     7       'YSID      ZUCL'/
     8       'ZLSCP     ZLCL')
90011 FORMAT(' ')
90012 FORMAT(I2,I3,5X,6F10.5)
90013 FORMAT(' 0')
90014 FORMAT(3F10.5,F9.5)
21700 FORMAT(' 1',78X/
     1' 1 6      NONCIRCULAR CROSS SECTION                       ',21X/
     2'LELL     1ELLI  PIEC    BLCL       SID       LSCP         ',21X/
     3'UELL     2ELLI  PIEC     SID      TUCL       USCP         ',21X/
     4'LBSL     3LINE  PIEC    BLSL       LSLCP
     5LELL',11X/
     6'LSSL     4LINE  PIEC     LSLCP     SSL               LELL',21X/
     7'USSL     5LINE  PIEC     SSL       USLCP
     8UELL',11X/
     9'UTSL     6LINE  PIEC     USLCP    TUSL               UELL',21X/
     A       ' 1        MAPAX')
21900 FORMAT('YUSL      YUCL',66X/
     1       'YLSL      YUCL',66X/
     6       'YUSLCP    YSSL',66X/
     7       'YLSLCP    YSSL',66X/
     A       'ZUSLCP    ZUSL',66X/
     B       'ZLSLCP    ZLSL',66X/
     C       'ZSSL      ZSID',66X)
      END
C***********************************************************************
      SUBROUTINE LIST
C---------------------
C LIST QUICK INPUT FILE
C---------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(1U),X2(10),Z1(10),Z2(10),T1(10),T2(10),
```

Table 8. Continued

```
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PI0180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CAND,TI,T
      IF(IF.EQ.IP)GO TO 9060
 9000 CONTINUE
      REWIND IF
      WRITE(IW,9010)IF
 9010 FORMAT('0LISTING OF QUICK INPUT DECK ON UNIT ',I2,':')
      WRITE(IW,2007)
 9020 READ(IF,90001,END=9040)IDEN
      WRITE(IW,2009)IDEN
 9030 FORMAT(' ',80A1)
      GO TO 9020
 9040 WRITE(IW,2007)
 9050 FORMAT(' ',132('.'))
      WRITE(IW,9055)
 9055 FORMAT('1')
 9060 RETURN
90001 FORMAT(80A1)
 2007 FORMAT(' ',82('-'))
 2009 FORMAT(' |',80A1,'|')
      END
C***********************************************************************
      SUBROUTINE SLCE
C---------------------
C COMPUTE SLICE GEOMETRY
C---------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PI0180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
```

Table 8. Continued

```
10000 CONTINUE
      IF(TYPE.NE.ASYM)GO TO 10006
C SET UP XQ AND ZQ FOR ASYM CASE
      NSEGP1=1
      XQ(1)=VB(NB(1)+1,1,1,1)
      A1=0.
      DO 10005 L=1,3
      I1=NB(L)+1
      DO 10005 I=1,I1
      A1=AMAX1(A1,VB(I,2,1,L))
10005 CONTINUE
      ZQ(1)=A1
10006 CONTINUE
C DETERMINE IF SLICES ARE PRESENT
      ISL=0
      DO 10010 I=1,10
      DO 10010 J=1,3
      DO 10010 L=1,3
      IF(V(I,J,L).GT.RINE)GO TO 10010
      ISL=1
      GO TO 10015
10010 CONTINUE
      IF(TYPE.EQ.ASYM.AND.ISL.EQ.0)GO TO 10015
      IF(ISL.EQ.0)RETURN
C DETERMINE NUMBER OF SEGMENTS IN EACH SLICE NS(L)=NP(L)-1
10015 DO 10020 L=1,3
      NP(L)=0
      DO 10020 I=1,10
      DO 10020 J=1,3
      IF(V(I,J,L).LT.RINF)NP(L)=I
10020 CONTINUE
      DO 10030 L=1,3
      NS(L)=NP(L)-1
10030 CONTINUE
C CHECK INPUT FOR SUFFICIENCY
      DO 10060 L=1,3
      I1=NP(L)
      IF(I1.EQ.0)GO TO 10060
      DO 10055 I=1,I1
      I2=0
      DO 10040 J=1,3
      IF(V(I,J,L).LT.RINF)I2=I2+1
10040 CONTINUE
      IF(I2.GE.2)GO TO 10055
      ISTOP=1
      WRITE(IW,10050)L,I
10050 FORMAT(' COND CODE 0050: SLICE',I2,', POINT',I3,', DATA INSUFFICIE
     1NT')
10055 CONTINUE
10060 CONTINUE
      IF(ISTOP.EQ.0)GO TO 10070
      WRITE(IW,2035)
 2035 FORMAT(' EXECUTION HALTED')
      STOP 0050
C COMPUTE REMAINING SLICE GEOMETRY
10070 DO 10160 L=1,3
```

Table 8. Continued

```
      I1=NP(L)
      IF(I1.EQ.0)GO TO 10160
      DO 10150 I=2,I1
      IM1=I-1
      DO 10080 J=1,3
      IF(V(I,J,L).GT.RINF)GO TO(10100,10130,10140),J
10080 CONTINUE
      A1=(V(I,2,L)-V(IM1,2,L))/(V(I,1,L)-V(IM1,1,L))
      A1=ATAN(A1)/PIO180
      IF(ABS(A1-V(I,3,L)).LE.ERRMAX)GO TO 10150
      ISTOP=1
      WRITE(IW,10090)L,I,ERRMAX
10090 FORMAT(' COND CODE 0090: SLICE',I2,', POINT',I3,', REDUNDANT INPUT
     1 DATA NOT SELF CONSISTENT TO WITHIN ',1PE15.7)
      GO TO 10150
10100 IF(V(I,3,L).EQ.0.)GO TO 10110
      V(I,1,L)=(V(I,2,L)-V(IM1,2,L))/TAN(PIO180*V(I,3,L))+V(IM1,1,L)
      GO TO 10150
10110 WRITE(IW,10120)L,I
10120 FORMAT(' COND CODE 0120: SLICE',I2,', POINT',I3,', X MUST BE INPUT
     1 WHEN SLOPE IS ZERO')
      WRITE(IW,2035)
      STOP 0120
10130 V(I,2,L)=V(IM1,2,L)+(V(I,1,L)-V(IM1,1,L))*TAN(PIO180*V(I,3,L))
      GO TO 10150
10140 V(I,3,L)=ATAN((V(I,2,L)-V(IM1,2,L))/(V(I,1,L)-V(IM1,1,L)))/PIO180
10150 CONTINUE
10160 CONTINUE
      IF(ISTOP.EQ.0)GO TO 10170
      WRITE(IW,2035)
      STOP 0090
C EXTRAPOLATE BEGINNING OF FIRST SEGMENT TO X=0
10170 DO 10180 L=1,3
      I1=NP(L)
      IF(I1.EQ.0)GO TO 10180
      IF(V(1,1,L).LE.0.) GO TO 10180
      V(1,2,L)=V(1,2,L)-V(1,1,L)*TAN(PIO180*V(2,3,L))
      V(1,1,L)=0.
10180 CONTINUE
C EXTRAPOLATE END OF LAST SEGMENT TO XQ(NSEGP1)
      DO 10190 L=1,3
      I1=NP(L)
      IF(I1.EQ.0)GO TO 10190
      IF(V(I1,1,L).GE.XQ(NSEGP1))GO TO 10190
      V(I1,2,L)=V(I1,2,L)+(XQ(NSEGP1)-V(I1,1,L))*TAN(PIO180*V(I1,3,L))
      V(I1,1,L)=XQ(NSEGP1)
10190 CONTINUE
C PLACE NON EXISTENT SLICES OUTSIDE OF VEHICLE
      A1=0.
      DO 10200 I=1,NSEGP1
      A1=AMAX1(A1,ABS(ZQ(I)))
10200 CONTINUE
      A1=2.*A1
      DO 10210 L=1,3
      IF(NP(L).NE.0)GO TO 10210
      NP(L)=2
```

Table 8. Continued

```
      NS(L)=1
      V(1,1,L)=0.
      V(1,2,L)=A1
      V(2,1,L)=XQ(NSEGP1)
      V(2,2,L)=A1
      V(1,3,L)=0.
      V(2,3,L)=0.
10210 CONTINUE
C BE SURE V(1,3,L) IS DEFINED
      DO 10220 L=1,3
      IF(V(1,3,L).GT.RINF)V(1,3,L)=V(2,3,L)
10220 CONTINUE
C PRINT RESULTS OF SLICE GEOMETRY CALCULATIONS
      WRITE(IW,10230)
10230 FORMAT('0SLICE GEOMETRY:'/'0',21X,'UPPER',39X,'LOWER',39X,'SIDE '
     1/ 9X,'X',13X,'Z' ,13X,'T',15X,'X',13X,'Z',13X,'T',15X,'X',13X,
     2'Y',13X,'T'/' ',132('-'))
      DO 10250 I=1,10
      WRITE(IW,10240)I,((V(I,J,L),J=1,3),L=1,3)
10240 FORMAT(' ',I2,2(1P3E14.6,2X),1P3E14.6)
10250 CONTINUE
C CHECK VALIDITY OF COORDINATES
      DO 10280 L=1,3
      I1=NP(L)
      DO 10270 I=1,I1
      IF(V(I,2,L).GE.0.)GO TO 10270
      ISTOP=1
      WRITE(IW,10260)L,I
10260 FORMAT(' COND CODE 0260: SLICE',I2,', POINT',I3,', Z VALUE < 0. SL
     1ICES MAY NOT CROSS VEHICLE CL.')
10270 CONTINUE
10280 CONTINUE
      IF(ISTOP.EQ.0)GO TO 10290
      WRITE(IW,2035)
      STOP 0260
C RESIGN LSL FOR QUICK INPUT
10290 I1=NP(2)
      IF(TYPE.EQ.ASYM)RETURN
      DO 10300 I=1,I1
      DO 10300 J=2,3
      V(I,J,2)=-V(I,J,2)
10300 CONTINUE
      RETURN
      END
C***********************************************************************
      SUBROUTINE RFSL
C-----------------------------------------------------------------
C INSERT ROUNDS OR FILLETS BETWEEN SEQUENTIAL SLICING PLANES
C-----------------------------------------------------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
```

Table 8. Continued

```
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
11000 CONTINUE
      ISKIP=0
      IF(ISL.EQ.0)RETURN
C DETERMINE IF ROUNDS OR FILLETS ARE PRESENT
      IRFSL =0
      DO 11005 L=1,3
      DO 11005 I=1,10
      IF(RHOSL(I,L).GT.RINF)GO TO 11005
      IRFSL=1
      GO TO 11006
11005 CONTINUE
      RETURN
11006 WRITE(IW,11007)
11007 FORMAT('0SLICE ROUND/FILLET CENTERS AND INTERSECTIONS:'
     1      /'0', 9X,'BI',11X,'BJ',11X,'A3',11X,'A4',11X,'X1',11X,'Z1',
     1            11X,'T1',11X,'X2',11X,'Z2',11X,'T2'/' ',
     2 132('-'))
C COMPUTE ROUND/FILLET CENTERS AND INTERSECTIONS
      DO 11130 L=1,3
      DO 11030 I=2,9
      IP1=I+1
      IF(RHOSL(I,L).GT.RINF)GO TO 11030
      ISKIP=1
      IF(ABS(V(I,3,L)-V(IP1,3,L)).GE.ERRMAX)GO TO 11020
      WRITE(IW,11010)I,IP1
11010 FORMAT(' COND CODE 1010: FILLETS/ROUNDS MAY NOT BE INSERTED BETWEE
     1N LINE SEGMENTS WITH SAME SLOPE, SEGMENTS',2I3)
      WRITE(IW,2035)
 2035 FORMAT(' EXECUTION HALTED')
      STOP 1010
11020 S=1.
      IF(V(IP1,3,L).GT.V(I,3,L))S=-1.
      MI=TAN(PIO180*V(I,3,L))
      MJ=TAN(PIO180*V(IP1,3,L))
      A1=V(I,2,L)-MI*V(I,1,L)
      A2=V(IP1,2,L)-MJ*V(IP1,1,L)
      BI=A1-S*RHOSL(I,L)/COS(PIO180*V(I,3,L))
      BJ=A2-S*RHOSL(I,L)/COS(PIO180*V(IP1,3,L))
      A3=(MJ*BI-MI*BJ)/(MJ-MI)
      A4=-(BJ-BI)/(MJ-MI)
      X1(I)=A4-S*RHOSL(I,L)*SIN(PIO180*V(I,3,L))
      X2(I)=A4-S*RHOSL(I,L)*SIN(PIO180*V(IP1,3,L))
      Z1(I)=A3+S*RHOSL(I,L)*CUS(PIO180*V(I,3,L))
      Z2(I)=A3+S*RHOSL(I,L)*COS(PIO180*V(IP1,3,L))
      T1(I)=V(I,3,L)
      T2(I)=V(IP1,3,L)
```

Table 8. Continued

```
      WRITE(IW,11025)L,I,BI,BJ,A3,A4,X1(I),Z1(I),T1(I),X2(I),Z2(I),T2(I)
11025 FORMAT(1X,2I1,1P10E13.5)
11030 CONTINUE
C INSERT ROUNDS/FILLETS INTO QUICK ARRAYS
      DO 11035 I=1,9
      IP1=I+1
      RHOSL(I,L)=RHOSL(IP1,L)
      X1(I)=X1(IP1)
      Z1(I)=Z1(IP1)
      T1(I)=T1(IP1)
      X2(I)=X2(IP1)
      Z2(I)=Z2(IP1)
      T2(I)=T2(IP1)
11035 CONTINUE
      RHOSL(10,L)=1.E70
      J=0
      DO 11110 I=1,10
      J=J+1
      IF(RHOSL(I,L).GT.RINF)GO TO 11110
      J=J+1
      NS(L)=NS(L)+1
      NP(L)=NP(L)+1
      IF(NP(L).LE.10)GO TO 11050
      WRITE(IW,11040)I
11040 FORMAT(' COND CODE 1040: INSERTION OF ROUND/FILLET',I3,' EXCEEDS D
     1IMENSIONS (10)')
      WRITE(IW,2035)
      STOP 1040
11050 N=11
11060 N=N-1
      NM1=N-1
      IF(N.LE.J)GO TO 11080
      DO 11070 I2=1,3
11070 V(N,I2,L)=V(NM1,I2,L)
      CRVSL(N,L)=CRVSL(NM1,L)
      GO TO 11060
11080 CONTINUE
      V(J,1,L)=X1(I)
      V(J,2,L)=Z1(I)
      V(J,3,L)=T1(I)
      CRVSL(J+1,L)=CRVSL(J,L)
      V(J+1,1,L)=X2(I)
      V(J+1,2,L)=Z2(I)
      V(J+1,3,L)=T2(I)
      CRVSL(J,L)=ELLX
11110 CONTINUE
      WRITE(IW,11120)
11120 FORMAT(' ')
11130 CONTINUE
C PRINT FINAL QUICK ARRAYS
      WRITE(IW,10230)
10230 FORMAT('0FINAL QUICK ARRAYS FOR ROUNDED/FILLETTED SLICES:',
     1                        /'0',21X,'UPPER',39X,'LOWER',39X,'SIDE '
     1/ 9X,'X',13X,'Z' ,13X,'T',15X,'X',13X,'Z',13X,'T',15X,'X',13X,
     2'Y',13X,'T'/' ',132('-'))
      DO 10250 I=1,10
```

Table 8. Continued

```
      WRITE(IW,10240)I,((V(I,J,L),J=1,3),L=1,3)
10240 FORMAT(' ',I2,2(1P3E14.6,2X),1P3E14.6)
10250 CONTINUE
      RETURN
      END
C**********************************************************************
      SUBROUTINE COPY
C-----------------------------------
C COPY DATA INTO VB ARRAY FOR QK3D
C-----------------------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
      DIMENSION E(10,7)
      EQUIVALENCE (E(1,1),X (1))
     1            ,(E(1,2),Z (1))
     2            ,(E(1,3),TH(1))
     3            ,(E(1,4),H (1))
     4            ,(E(1,5),K (1))
     5            ,(E(1,6),R (1))
     6            ,(E(1,7),B (1))
      DATA RINF1/1.E70/
      LOGICAL*1 F/'F'/
12000 CONTINUE
C                1     2     3     4     5     6     7     8
      GO TO(12010,12010,12010,12050,12050,12050,12080,12080
     1),LB
C LB=1 ZUCL, LB=2 ZLCL, LB=3 YSID
12010 NB(LB)=NSEG
      DO 12020 I=1,10
      VB(I,1,1,LB)=XQ(I)
      VB(I,2,1,LB)=ZQ(I)
      VB(I,3,1,LB)=TQ(I,1)
      VB(I,3,2,LB)=TQ(I,2)
      CRVB (I,LB)=CURVE(I)
12020 CONTINUE
      DO 12040 I=1,10
      XQ(I)=RINF1
      ZQ(I)=RINF1
      TQ(I,1)=RINF1
      TQ(I,2)=RINF1
      DO 12030 J=1,7
```

Table 8. Continued

```
      E(I,J)=RINF1
12030 CONTINUE
      SLOPE(I)=F
      RHO(I)=RINF1
      X1(I)=RINF1
      X2(I)=RINF1
      Z1(I)=RINF1
      Z2(I)=RINF1
      T1(I)=RINF1
      T2(I)=RINF1
      KSI(I)=RINF1
      ETA(I)=RINF1
      CURVE(I)=BLNK
      DIR(I)=OPDN
12040 CONTINUE
      RETURN
C LB=4 ZUSL, LB=5 ZLSL, LB=5 YSSL
12050 I1=LB-3
      NB(LB)=NS(I1)
      DO 12070 I=1,10
      DO 12060 J=1,3
      VB(I,J,1,LB)=V(I,J,I1)
      CRVB(I,LB)=CRVSL(I,I1)
12060 CONTINUE
      IF(I.LT.10)VB(I,3,2,LB)=V(I+1,3,I1)
12070 CONTINUE
      RETURN
12080 DO 12090 I=1,10
      VB(I,1,1,7)=XSID(I)
      VB(I,2,1,7)=ZSID(I)
      VB(I,1,1,8)=XMAPAX(I)
      VB(I,2,1,8)=ZMAPAX(I)
12090 CONTINUE
      RETURN
      END
C*****************************************************************
      SUBROUTINE BLCK
C-----------------------------------------
C CHECK ALL BODY LINE COORDINATES FOR VALIDITY
C COMPUTE DEFAULT VALUES FOR ZSID AND ZMAPAX
C-----------------------------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
```

Table 8. Continued

```
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
13000 CONTINUE
C ZSID AND ZMAPAX DEFAULT
      DO 13060 L=7,8
      NB(L)=0
      DO 13010 I=1,10
      IF(VB(I,1,1,L).LT.RINF.AND.VB(I,2,1,L).LT.RINF)NB(L)=I
13010 CONTINUE
      NB(L)=NB(L)-1
      IF(NB(L).GT.0)GO TO 13060
      IF(NB(L).EQ.-1)GO TO 13030
      WRITE(IW,13020)L
13020 FORMAT(' COND CODE 3020: BODY LINE',I2,' REQUIRES X AND Z VALUES F
     1OR TWO OR MORE POINTS')
      WRITE(IW,2035)
 2035 FORMAT(' EXECUTION HALTED')
      STOP 3020
13030 IF(L.NE.7)GO TO 13040
      NB(7)=1
      VB(1,1,1,7)=0.
      VB(1,2,1,7)=0.
      VB(2,1,1,7)=XQ(NSEGP1)
      VB(2,2,1,7)=0.
      GO TO 13060
13040 I1=NB(7)+1
      NB(8)=NB(7)
      DO 13050 I=1,I1
      DO 13050 J=1,2
      VB(I,J,1,8)=VB(I,J,1,7)
13050 CONTINUE
13060 CONTINUE
C VALIDITY CHECK
      A1=VB(NB(1)+1,1,1,1)
      DO 13080 L=2,8
      I1=NB(L)+1
      IF(ABS(VB(I1,1,1,L)-A1).LE.ERRMAX)GO TO 13080
      IF(VB(I1,1,1,L).GE.A1.AND.L.GE.4)GO TO 13080
      WRITE(IW,13070)ERRMAX,L
13070 FORMAT(' COND CODE 3070: ALL BODY LINES MUST BE THE SAME LENGTH TO
     1 WITHIN ERRMAX=',1PE14.7,' BL',I2)
      WRITE(IW,2035)
      STOP 3070
13080 CONTINUE
      DO 13100 L=1,8
      IF(ABS(VB(1,1,1,L)).LE.ERRMAX)GO TO 13100
      WRITE(IW,13090)ERRMAX,L,VB(1,1,1,L)
13090 FORMAT(' COND CODE 3090: FIRST X OF EACH BODY LINE MUST BE ZERO (T
     1O WITHIN ERRMAX=',1PE14.7,') BL',I2,' X=',1PE14.7)
      WRITE(IW,2035)
      STOP 3090
13100 CONTINUE
C RESIGN ZLCL AND ZLSL
      DO 13110 L=2,5,3
      I1=NB(L)+1
      DO 13110 I=1,I1
```

Table 8. Continued

```
      DO 13110 J=2,3
      DO 13110 N=1,2
      IF(VB(I,J,N,L).GT.RINF)GO TO 13110
      VB(I,J,N,L)=-VB(I,J,N,L)
13110 CONTINUE
C SET UP ANGLES FOR ZMAPAX AND ZSID
      DO 14026 L=7,8
      I1=NB(L)
      DO 14025 I=1,I1
      A1=ATAN((VB(I+1,2,1,L)-VB(I,2,1,L))/(VB(I+1,1,1,L)-VB(I,1,1,L)))
      A1=A1/PIO180
      VB(I,3,1,L)=A1
      VB(I,3,2,L)=A1
14025 CONTINUE
14026 CONTINUE
      RETURN
      END
C***********************************************************************
      SUBROUTINE QK3D
C-----------------------------
C SET UP QUICK MODEL FOR 3D CASE
C-----------------------------
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),R¬OSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PIO180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
14000 CONTINUE
      IF(IF.NE.IP)REWIND IF
C CS MODELS
      WRITE(IF,90001)IDEN
      WRITE(IF,21700)
      I1=1
      WRITE(IF,90004)I1,I1,ZERO,VB(NB(1)+1,1,1,1)
C BL MODELS
      WRITE(IF,90005)YUCL
      WRITE(IF,90006)I1,KRV(1),KV(1)
      WRITE(IF,90007)ZERO,ZERO,VB(NB(1)+1,1,1,1),ZERO,ZERO,ZERO
      WRITE(IF,90009)
      DO 14020 L=1,8
      I1=NB(L)
      WRITE(IF,90005)BLB(L)
      DO 14010 I=1,I1
      IP1=I+1
```

Table 8. Continued

```
      I2=1
      IF(CRVB(I,L).EQ.ELLX)I2=2
      WRITE(IF,90006)I,CRVB(I,L),KV(I2)
      WRITE(IF,90007)VB(I,1,1,L),VB(I,2,1,L),VB(IP1,1,1,L),VB(IP1,2,1,L)
     1,VB(I,3,1,L),VB(I,3,2,L)
14010 CONTINUE
      WRITE(IF,90009)
14020 CONTINUE
      WRITE(IF,21900)
      WRITE(IF,90011)
      IF(IPRINT.EQ.0)GO TO 14030
C SET UP CARDS TO MAKE QUICK EXERCISE MODEL
      DX=.001
      A1=DX
      A2=VB(NB(1)+1,1,1,1)-DX
      A3=(A2-A1)*.01
      A2=A2+2.*DX
      I1=1
      I2=2
      WRITE(IF,90012)I1,I2,A1,A2,A3
14030 WRITE(IF,90013)
      RETURN
90001 FORMAT(80A1)
90004 FORMAT(2I2,6X,2F10.5)
90005 FORMAT(A4)
90006 FORMAT(I2,1X,A4,'   PIEC  ',A4)
90007 FORMAT(3F10.5,2(F9.5,'A'),F10.5)
90009 FORMAT('-1')
90011 FORMAT(' ')
90012 FORMAT(I2,I3,5X,6F10.5)
90013 FORMAT(' 0')
90014 FORMAT(3F10.5,F9.5)
21700 FORMAT(' 1',78X/
     1' 1 6      NONCIRCULAR CROSS SECTION                        ',21X/
     2'LELL     1ELLI  PIEC    8LCL       SID       LSCP          ',21X/
     3'UELL     2ELLI  PIEC     SID      TUCL       USCP          ',21X/
     4'LBSL     3LINE  PIEC    BLSL       LSLCP
     5LELL',11X/
     6'LSSL     4LINE  PIEC     LSLCP     SSL                 LELL',21X/
     7'USSL     5LINE  PIEC     SSL       USLCP
     8UELL',11X/
     9'UTSL     6LINE  PIEC     USLCP    TUSL                 UELL',21X/
     A      ' 1        MAP   ')
21900 FORMAT('YUSL      YUCL',66X/
     1       'YLSL      YUCL',66X/
     2       'YLCL      YUCL',66X/
     3       'YMAP      YUCL',66X/
     4       'YUSCP     YSID',66X/
     5       'YLSCP     YSID',66X/
     6       'YUSLCP    YSSL',66X/
     7       'YLSLCP    YSSL',66X/
     8       'ZUSCP     ZUCL',66X/
     9       'ZLSCP     ZLCL',66X/
     A       'ZUSLCP    ZUSL',66X/
     B       'ZLSLCP    ZLSL',66X/
     C       'ZSSL      ZSID',66X)
```

Table 8. Concluded

```
      END
C************************************************************************
      BLOCKDATA
      COMMON /RA/X(10),Z(10),TH(10),H(10),K(10),R(10),B(10),M(10),
     1 CURVE(10),DIR(10),X1(10),X2(10),Z1(10),Z2(10),T1(10),T2(10),
     2 KSI(10),ETA(10),RHO(10),XQ(11),ZQ(11),TQ(11,2),KRV(3),CL(2),
     3 KV(2),V(10,3,3),SL(3),RHOSL(10,3),CRVSL(10,3)
     4,VB(10,3,2,8),CRVB (10,8),XMAPAX(10),ZMAPAX(10),XSID(10),ZSID(10),
     5 BLB(8)
      COMMON /IA/ICASE(10),JCASE(30),NCASE(7),NP(3),NS(3),NB(8)
      COMMON /RS/KI,MI,KJ,MJ,MIJ,LINE,BLNK,RINF,ERRMAX,OPDN,ELLX,
     1 YUCL,ZERO,FORE,AFTE,TOPS,BOTS,OPUP,OGIV,PI0180,XJ,ZJ,XI,ZI,HI,
     2 RI,BI,THI,BJ,A1,A2,A3,BIJ,THJ,S,THIJ,SIJ,AI,HJ,RJ,A4,DX
     3,TYPE,ASYM,UPFO,UPAF,UPTO,UPBO,DNFO,DNAF,DNTO,DNBO
      COMMON /IS/IR,IW,IF,IP,IPRINT,NSEG,NSEGP1,ISTOP,JSTOP,KSTOP,
     1 I,J,L,IP1,LM1,I1,NPTS,I2,ISL,LB,N,NM1,I3
      COMMON /LA/SLOPE(10),IDEN(80),CARD(80)
      COMMON /LS/TI,T
      REAL K,M,KSI,KI,MI,KJ,MJ,MIJ,LINE,KV,KRV
      LOGICAL*1 SLOPE,IDEN,CARD,TI,T
      DATA KRV/'LINE','OGIV','ELLX'/,LINE/'LINE'/,CL/'ZUCL','ZLCL'/,
     1 KV/'KV5 ','KV0 '/,SLOPE/10*'F'/,T/'T'/
      DATA RHO,X1,X2,Z1,Z2,T1,T2,KSI,ETA/90*1.E70/
      DATA X,Z,      R,H,K,M,B,TH/ 80*1.E70/
      DATA                CURVE/10*'    '/,BLNK/'    '/,RINF/1.E69/,
     1 ERRMAX/1.E-5/,DIR/10*'OPDN'/,OPDN/'OPDN'/,ELLX/'ELLX'/,
     2 YUCL/'YUCL'/,ZERO/0./,FORE/ 'FORE'/,AFTE/'AFTE'/,TOPS/'TOPS'/,
     3 BOTS/'BOTS'/,OPDN/'OPDN'/,OGIV/'OGIV'/,OPUP/'OPUP'/
      DATA XQ,ZQ,TQ/44*1.E70/
      DATA V/ 90*1.E70/,SL/'ZUSL','ZLSL','YSSL'/
      DATA RHOSL/30*1.E70/,CRVSL/30*'LINE'/
      DATA VB/480*1.E70/,CRVB /80*'LINE'/,TYPE/'SYMM'/,ASYM/'ASYM'/
      DATA BLB/'ZUCL','ZLCL','YSID','ZUSL','ZLSL','YSSL','ZSID','ZMAP'/
      DATA XMAPAX,ZMAPAX,XSID,ZSID/40*1.E70/
      DATA UPFO , UPAF , UPTO , UPBO , DNFO , DNAF , DNTO , DNBO
     1   /'UPFO','UPAF','UPTO','UPBO','DNFO','DNAF','DNTO','DNBO'/
      DATA JCASE/582,774,834,526,624,006,010,012,136,590,
     1           626,628,632,014,138,140,142,638,832,262,
     2           838,
     -9*0/
      DATA NCASE/  1,  2,  3,  4,  6,
     1             7,  8           /
      END
```

Table 9. Subroutine Description

Subroutine Name	Function
Main Program	Overall program control
INIT	Initializes certain program variables
INPT	Reads and prints input data
CASE	Determines the case number ICASE(I) of the input option for each segment (see Table 2)
SEGE	Determines the equation of each segment and computes certain intersection information
INTR	Determines intersection points of successive segments for which only segment equations are known to that point in code
QARY	Assembles coordinate and slope data in convenient arrays for building QUICK input deck (QUCK)
PRNT	Prints the QUICK input arrays assembled by QARY
RϕFL	Determines the equations of rounds and fillets and computes intersections with body line segments, prints results
QUCK	Builds the QUICK input deck for the symmetrical case with asymmetrical slices
LIST	Prints the QUICK input deck (if unit IF is not the card punch) for checking by user
SLCE	Performs geometry computation for slicing planes, prints results
RFSL	Determines the equations of rounds and fillets and computes intersections with slicing plane segments, prints results
CϕPY	Assembles coordinate and slope data in convenient arrays for building QUICK input deck for asymmetrical case (QK3D)

Table 9. Concluded

BLCK	Computes default values of certain body line coordinates and checks validity of previously computed coordinates for asymmetrical case
QK3D	Builds the QUICK input deck for the asymmetrical case with slices
BLϕCKDATA	Initializes certain program constants

Table 10. Variable Description

Variable Name	Type*	Description
B	RA	Z intercept of straight body line segment
H	RA	X coordinate of ogive center
I	IS	Intersection subscript
J	IS	Scratch integer
K	RA	Z coordinate of ogive center
L	IS	Scratch integer
M	RA	Slope of body line segment
N	IS	Scratch integer
R	RA	Radius of ogival segment
S	RS	Variable used in selection from multiple intersections, S=±1.
T	LS	='T'
V(I,J,L)	RA	Slice geometry array: J=1, X coordinate; J=2, Z (or Y) coordinate; J=3, slope angle (deg); L=1, upper slice; L=2, lower slice; L=3, side slice
X	RA	Input array for X coordinate of body lines
Z	RA	Input array for Z (or Y) coordinate of body lines
Ai	RS	Scratch variables i=1,2,...
BI	RS	Z intercept of Ith segment
BJ	RS	Z intercept of I-1st segment
CL	RA	Body line names used in symmetrical case to assemble QUICK input deck

Table 10. Continued

DX	RS	Round-off compensation variable
HI	RS	X coordinate of ogive center for Ith segment
HJ	RS	X coordinate of ogive center for I-1st segment
IF	IS	Unit number (data set reference number) of QUICK file written by QUCK and QK3D
IP	IS	Unit number of facility card punch
IR	IS	Unit number for input data
IW	IS	Unit number for facility printer
Ii	IS	Scratch integers, i=1,2,...
KI	RS	Z coordinate of ogive enter for Ith segment
KJ	RS	Z coordinate of ogive center for I-1st segment
KV	RA	Segment type (line or ogive)
LB	IS	Overall $D\phi$ variable for asymmetrical case, LB=1, upper centerline; LB=2, lower centerline; LB=3, side (see Fig. 7)
MI	RS	Slope of Ith segment (straight line)
MJ	RS	Slope of I-1st segment (straight line)
NB(LB)	IA	Number of segments in the body line segments in asymmetrical case
NP	IA	NS+1
NS(L)	IA	Number of segments in the three slicing planes, L defined as for V above
RI	RS	Radius of ogive for Ith segment
RJ	RS	Radius of ogive for I-1st segment

Table 10. Continued

SL	RA	Body line names for slicing planes
TH	RA	Input array for local slopes of body lines at end of segment
TI	LS	=SLϕPE(I)
TQ(I,J)	RA	Local segment slope (degrees): J=1, at beginning of Ith segment; J=2 at end of Ith segment; input to subroutine QUCK
Ti	RA	Slope (degrees) of fillet or round: i=1, beginning of fillet; i=2, end of fillet

VB(I,J,N,L)	RA	Coordinates and slopes of the eight body lines of the asymmetrical case:
		J=1, X coordinate
		J=2, Z coordinate
		J=3, slope (deg); N=1, beginning of Ith segment, N=2 end of Ith segment (N=2 used only with J=3)
		L=1 Upper centerline (ZUCL)
		L=2 Lower centerline (ZLCL)
		L=3 Side (YSID)
		L=4 Upper slicing plane (ZUSL)
		L=5 Lower slicing plane (ZLSL)
		L=6 Side slicing plane (YSSL)
		L=7 Side (ZSID)
		L=8 Map axis (ZMAPAX)
		(see Fig. 8)

XI	RS	X coordinate of end of Ith segment
XJ	RS	X coordinate of end of I-1st segment
XQ	RA	X coordinate of beginning of Ith segment (I=1,NSEG) and end of last segment (I=NSEGP1), input to QUCK
Xi	RA	X coordinate at intersections of fillet or round with body lines, i as above in Ti
ZI	RS	Z coordinate at end of Ith segment
ZJ	RS	Z coordinate at end of I-1st segment

Table 10. Continued

ZQ	RA	Z coordinate of beginning of Ith segment (I=1,NSEG) and end of last segment (I=NSEGP1), input to QUCK
Zi	RA	Z coordinate at intersections of fillet or round with body lines, i as above in Ti
BIJ	RS	Z intercept
BLB(L)	RA	Body line names for QK3D, L as above for VB
DIR	RA	Selection indicator variable for multiple root intersections
ETA	RA	Z coordinate of round or fillet center at end of Ith segment
IP1	IS	Scratch integer, =I+1
ISL	IS	Slice indicator: ISL=0 no slices ISL=1 slices present
KRV	RA	Segment type, line or ogive
KSI	RA	X coordinate of round or fillet center at end of Ith segment
LM1	IS	scratch integer, =L-1
MIJ	RS	Tangent of straight line slope angle
NM1	IS	Scratch integer, =N-1
RHϕ	RA	Input variable for radius of round or fillet at end of Ith segment
THI	RS	Slope (deg) at end of Ith segment
THJ	RS	Slope (deg) at end of I-1st segment
AFTE	RA	='AFTE', possible DIR value (aft solution)
ASYM	RA	='ASYM', possible TYPE value
BLNK	RS	= 'ƀƀƀƀ' (four blank characters)
BϕTS	RS	='BϕTS', possible DIR value (bottom solution)

Table 10. Continued

CARD(80)	LA	Card image array
CRVB(I,L)	RA	Curve type indicator for QUICK, CRVB ='LINE' or 'ELLX'
ELLX	RS	='ELLX'
FϕRE	RS	='FϕRE', possible DIR value (fore solution)
IDEN(80)	LA	Input variable for identification data
LINE	RS	='LINE', possible CURVE value
NPTS	IS	Number of stations at which QUICK will print out geometry data if IPRINT≠0
NSEG	IS	Number of segments for symmetrical case
ϕGIV	RS	='ϕGIV', possible CURVE value
ϕPDN	RS	='ϕPDN', possible DIR value (open down)
ϕPUP	RS	='ϕPUP', possible DIR value, (open up)
RINF	RS	Real infinity, = 1.E69
THIJ	RS	Straight-line slope (deg)
TLSL TSSL TUSL	RA	Input arrays for local slopes of the lower slice, side slice, and upper slice segments, respectively
XLSL XSSL XUSL	RA	Input arrays of X coordinate of end of lower, side, and upper slicing plane segments, respectively
ZLSL YSSL ZUSL	RA	Input arrays of Z coordinate of end of lower, side, and upper slicing plane segments, respectively
TϕPS	RS	='TϕPS', possible DIR value (top solution)
TYPE	RS	Type body: TYPE ='SYMM', symmetrical ='ASYM', assymetrical.
XSID	RA	X coordinate of side body line (Z)

Table 10. Continued

YUCL	RS	'YUCL'
ZERϕ	RS	Real zero = 0.0
ZSID	RA	Z coordinate of side body line
CRVSL(I,L)	RA	Curve type (='LINE' or 'ELLX') for slicing planes, L as in V above
CURVE	RA	Curve type for Ith segment, ='LINE' or 'ϕGIV'
ICASE	IA	Integer for identifying input option (see Table 2)
INPUT		Namelist name for input data
ISTϕP	IS	Error indicator to halt execution when set different from zero
JCASE	IA	Integer array used to determine ICASE (see Table 2)
JSTϕP	IS	Error indicator similar to ISTϕP
KSTϕP	IS	Error indicator similar to ISTϕP
NCASE	IA	ICASE values which input options may not follow ICASE = 5 or 9
RHϕLS	RA	Input array for lower slice fillet/round radii
RHϕSL(I,L)	RA	Array of fillet/round radii, L as in V above
RHϕSS	RA	Input array for side slice fillet/round radii
RHϕUS	RA	Input array for upper slice fillet/round radii
SLϕPE	LA	Variable to indicate Ith segment is to be tangent to end of I-1st segment
ERRMAX	RS	Computational error limit, = 5.E-5

Table 10. Concluded

IPRINT	IS	Print indicator: IPRINT≠0 causes KWIKNϕSE to generate cards to make QUICK exercise the completed math model
NSEGP1	IS	=NSEG+1
PIϕ180	RS	$\pi/180$
XMAPAX	RA	X coordinate of map axis on input
ZMAPAX	RA	Z coordinate of map axis on input

*R=real, I=integer, L=logical, A=array, S=scalar